

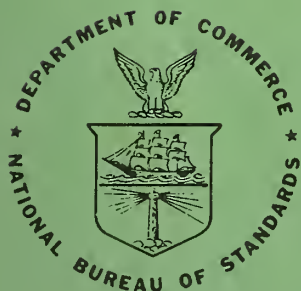
SEP 23 1968

**NBS**

**TECHNICAL NOTE**

432

**Connection Tables From Wiswesser  
Chemical Structure Notations —  
A Partial Algorithm**



**U.S. DEPARTMENT OF COMMERCE  
National Bureau of Standards**

## NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards<sup>1</sup> was established by an act of Congress March 3, 1901. Today, in addition to serving as the Nation's central measurement laboratory, the Bureau is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To this end the Bureau conducts research and provides central national services in three broad program areas and provides central national services in a fourth. These are: (1) basic measurements and standards, (2) materials measurements and standards, (3) technological measurements and standards, and (4) transfer of technology.

The Bureau comprises the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, and the Center for Radiation Research.

**THE INSTITUTE FOR BASIC STANDARDS** provides the central basis within the United States of a complete and consistent system of physical measurement, coordinates that system with the measurement systems of other nations, and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of an Office of Standard Reference Data and a group of divisions organized by the following areas of science and engineering:

Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic Physics—Cryogenics<sup>2</sup>—Radio Physics<sup>2</sup>—Radio Engineering<sup>2</sup>—Astrophysics<sup>2</sup>—Time and Frequency.<sup>2</sup>

**THE INSTITUTE FOR MATERIALS RESEARCH** conducts materials research leading to methods, standards of measurement, and data needed by industry, commerce, educational institutions, and government. The Institute also provides advisory and research services to other government agencies. The Institute consists of an Office of Standard Reference Materials and a group of divisions organized by the following areas of materials research:

Analytical Chemistry—Polymers—Metallurgy—Inorganic Materials—Physical Chemistry.

**THE INSTITUTE FOR APPLIED TECHNOLOGY** provides for the creation of appropriate opportunities for the use and application of technology within the Federal Government and within the civilian sector of American industry. The primary functions of the Institute may be broadly classified as programs relating to technological measurements and standards and techniques for the transfer of technology. The Institute consists of a Clearinghouse for Scientific and Technical Information,<sup>3</sup> a Center for Computer Sciences and Technology, and a group of technical divisions and offices organized by the following fields of technology:

Building Research—Electronic Instrumentation—Technical Analysis—Product Evaluation—Invention and Innovation—Weights and Measures—Engineering Standards—Vehicle Systems Research.

**THE CENTER FOR RADIATION RESEARCH** engages in research, measurement, and application of radiation to the solution of Bureau mission problems and the problems of other agencies and institutions. The Center for Radiation Research consists of the following divisions:

Reactor Radiation—Linac Radiation—Applied Radiation—Nuclear Radiation.

<sup>1</sup> Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D. C. 20234.

<sup>2</sup> Located at Boulder, Colorado 80302.

<sup>3</sup> Located at 5285 Port Royal Road, Springfield, Virginia 22151.

UNITED STATES DEPARTMENT OF COMMERCE  
C. R. Smith, Secretary  
NATIONAL BUREAU OF STANDARDS • A. V. Astin, Director



# TECHNICAL NOTE 432

ISSUED SEPTEMBER 1968

## **Connection Tables From Wiswesser Chemical Structure Notations — a Partial Algorithm**

George F. Fraction, Justin C. Walker,  
and Stephen J. Tauber

Systems Research and Development Division  
Center for Computer Sciences and Technology  
Institute for Applied Technology  
National Bureau of Standards  
Washington, D.C. 20234

NBS Technical Notes are designed to supplement the Bureau's regular publications program. They provide a means for making available scientific data that are of transient or limited interest. Technical Notes may be listed or referred to in the open literature.



CONNECTION TABLES FROM WISWESSER CHEMICAL STRUCTURE NOTATIONS---  
A PARTIAL ALGORITHM

George F. Fraction\*  
Justin C. Walker  
Stephen J. Tauber

An algorithm has been developed for transforming certain types of Wiswesser organic structure notations into connection tables. Acyclic and benzene structures are treated, and provision has been made for all of the types of contractions used by the Wiswesser notation system. A separate algorithm is presented for treating linearly fused ring aggregates. A syntax has been developed to describe those portions of Wiswesser notations which refer to non-benzene ring systems.

Key Words: acyclic, benzene, chemical structure notations, connection tables, contractions, ring system, syntax analysis, transformation algorithm, Wiswesser.

## INTRODUCTION

The purpose of our work with the Wiswesser notation was to make generally available an algorithm for generating connection tables from such structure notations. This work has had to be terminated before reaching its final objective and will not be resumed in the foreseeable future. We therefore wish to make our results available, although they are admittedly incomplete.

---

\*Currently employed by Information Services, Research, Development and Control, Eli Lilly & Co., Indianapolis, Indiana, 46206.



The algorithm here described will provide connection tables from the acyclic and benzene portions of a notation, and process all forms of contraction demanded by the rules. The known bugs in the algorithm are indicated, with suggestions for their elimination. Finally, our approach to handling non-benzene cyclic notations is described. This comprises a syntactic description of the portions of a notation enclosed in the L ... J or T ... J symbols. In addition, a separate program has been written to construct connection tables for linearly fused ring aggregates from the syntactically analyzed notations.

Hyde et al. have recently developed a program for a similar analysis of Wiswesser notations [1a]. The two approaches differ in several respects: Hyde et al. use notational symbols as the basic structural units, (e.g. Q, V, 3) whereas we use the individual atoms, bar hydrogen attached to carbon. They in turn indicate the number of hydrogen atoms attached to ring carbon atoms. The program they describe does not interpret substituent locants of benzene rings in terms of the individual ring atoms. They do not describe a method to handle the systematic contractions which the rules require. Another program for transforming Wiswesser notations also exists [1b] but it is not publicly available.

The programs herein described, with the exception of the one written in ALGOL-60, were implemented on the NBS PILOT computer facility, which has been decommissioned since this work. The language used was the assembly language, PEAP, providing for three-address instructions with operation code modifiers, sequence control register digits, and break-point control digit. The configuration of the system was 64K of 68-bit word core memory, magnetic tape, magnetic wire, punched tape, typewriter, and Dataphone in- and output and punched card and raster scanner input.

## 1. THE NOTATION

The notation rules underwent change during the time that this laboratory was working with them and therefore specific note had to be taken of the modifications.

The versions of the Wiswesser notation manual of October 1964 and of April 1965 were compared and in addition members of the Wiswesser notation committee were contacted directly on specific points.

The following specific changes were noted in the manual [2a, b]:

1. The rank of the slash (/) was shifted from between R and S to between hyphen (-) and (Ø). Cf. Rule 2.
2. The concept of "unshared bridge atoms" and a rule for their citation was added. Cf. Rule 31C.
3. A section on ring analysis was added. This does not affect the notation rules but does give suggestions useful in error-checking.
4. The section on chelate rings was eliminated.
5. A section was added which comments at length on redrawing structural formulae for easier encoding, with particular attention to uncrossing valence bonds.
6. Rules were added for enciphering ions, free radicals, and isotopic species. The specification of the location of charge, unpaired electron, or isotope relies on citing the number of the position within the symbol string of the symbol which refers to the atom in question. Cf. Rules 50-52.
7. Rules were added for citing several types of indeterminacies. Substituents can be cited as attached at an unknown point, to a cyclic atom or to an acyclic atom not otherwise specified, to a specific ring system, to a ring system between limits of two locants, to a specific side chain, or to a specific alkyl or alkylene chain. Symbols are defined for unspecified hydrocarbon moieties, for alkyl groups of specified size but unspecified structure, for generic halogen, and for generic metal. Monocoordinated Markush groups can be cited. Cf. Rules 53-63.

Correspondence with members of the Wiswesser notation committee disclosed no further extensions to the Wiswesser notation in the areas of Markush structures or partially indeterminate structures nor any rules to cover stereoisomerism, coordination complexes, or inorganic structures. There have, however, been proposals, not yet formalized in rules, which provide for:

1. The use of "D" in initiating the notation for a ring system which contains a donor-acceptor bond [3],
2. The use of "\_Ø" to signify a metallocene [4], and
3. The use of "C" for cis and "T" for trans [5a].

Rules have also recently been adopted to resolve alternatives among possible multiplier contractions [5b]. There evidently is still no provision to distinguish among cyclic double and triple bonds in certain circumstances. Thus, according to Rule 31 [2b], both cyclo-octa-1,3,5,7-tetraene (structure I) and cycloocta-1,3,5-triene-7-yne (structure II) would be enciphered as L8J.

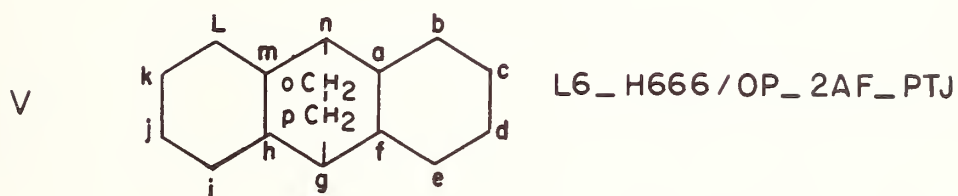
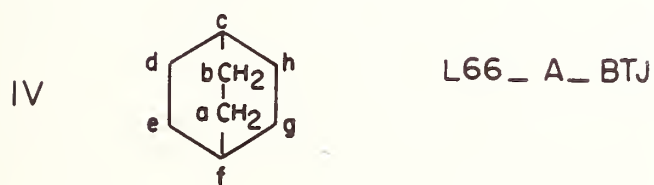
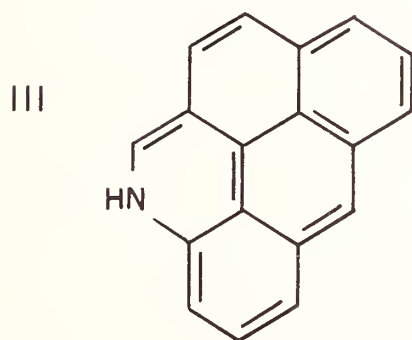
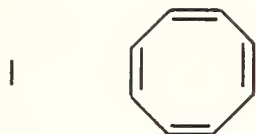
## 2. TRANSFORMATION ALGORITHM

The algorithm under development for transforming Wiswesser notations (WISCO) was being so designed as to yield connection tables compatible with those resulting from the computer routine developed for transforming Hayward notations [6]. The atoms would generally be listed in a different sequence depending on whether a Wiswesser or Hayward notation had provided the input, but the format of the connection tables resulting from the two transformations would be identical. An algorithm was essentially completed, programmed for, and partially debugged on the NBS PILOT computer facility to process benzene rings, acyclic portions, and multiplier contractions. The algorithm is shown at the stage it reached in Figures 1 through 19. The operation of this algorithm is highlighted in sections 2b and 2c below.

No algorithm usable for processing the general cyclic portions of Wiswesser notations has been devised. However, a linguistic structure of such cyclic portions was worked out as a basis for a transformation algorithm. Furthermore, an algorithm for analyzing a limited class of non-benzene cyclic notation portions was devised. The syntax for general cyclic portions and this limited algorithm are presented in section 2d.

No attempt was made in this phase of the work to deal with notations for salts, ions, free radicals, isotopic species, indeterminate locations, or generic substructures. For the present work a maximum coordination number of four has been assumed. Furthermore, it is suspected, although the algorithm was not developed far enough to verify this suspicion, that the processing of complex combinations of branched locants in ring system notations would require an effort disproportionate to the benefit to be derived from being able to deal with such rare cases. It appears to us sufficient to be able to construct connection tables from notations containing simple branched locants, and merely to shunt out those which contain either two branched locants from the same locant or a branched locant from a branched locant, i.e., those covered by Rule 39 [2b]. We also believe that it is sufficient to be able to process notations for a single ring of rings (Cf. Rule 44 [2b]); it is in any case not clear how Rule 44 is to be applied to structures with two or more rings of rings.





WISCO incorporates error checks both on the input notation and on the program itself. The errors are detected by looking for invalid situations and printing out an appropriate error message. For instance, if the character currently being examined is an "H" (hydrogen) or a "/" (slash mark) and the atom counter J has value 1, this indicates that the character initiates the notation and the cipher is rejected (except when the notation is HH). Another instance is a symbol for a multiple bond in an illegal context, e.g., UZ (i.e., =NH<sub>2</sub>, which, though chemically conceivable, would receive a different coding). When a symbol which stands for an atom set that demands a single bond connection to the rest of the compound occurs, the bond indicator (set to 2 or 3 when U or UU is encountered) is checked, and if the indicator is non-zero the notation is rejected.

Several ancillary routines were developed for use with WISCO. These were for convenience in implementation; they are not direct consequences of the logic evolved for transforming Wiswesser notations. Subroutine DEBUG (Cf. Figures 21 through 25) allows examination of system storage when a notation has been fully processed or when an error exit occurs. The programmer may then either look at storage or modify storage and re-enter WISCO at the beginning. The entry to WISCO is via subroutine INPUT, which reads a cipher one character at a time, checking each character for acceptability and storing it in a separate word of memory. (The 6-channel Flexowriter codes are converted upon input to 8-bit ASCII.) Two spaces in succession signify the end of the cipher. After a cipher has been read in, control passes to WISCO proper. Storage tables are set up not at assembly time but rather via subroutine GIMME (Cf. Figures 26 through 28). Storage is allocated pseudo-dynamically; i.e., storage is provided at run time, but if an overflow occurs then the storage is automatically augmented and the cipher is reprocessed afresh. GIMME is called with an argument SIZE. A value of zero for SIZE signals initialization for a new cipher. If SIZE has a "small" value, GIMME creates table storage of that size up to memory capacity and tells WISCO the table limits via SIZE. A "large" value of SIZE represents a table address and tells GIMME that a certain block of storage is being returned and causes this table to be erased from GIMME's records. "Large" and "small" values are defined by a constant in GIMME. GIMME will provide table storage of up to several thousand 68-bit words.

In the course of processing a notation there are several syntactic structures which require WISCO to employ recursive techniques. The recursion is facilitated by the use of pushdown stacks, i.e., last-in first-out storage, which are used for processing multiplicative contractions and branch symbols, as described in section 2a below.

## 2a. Multiplicative Contractions and Branch Symbols

Multiplicative contractions and chains of benzene rings necessitate recursive processing because multiplied strings may be nested (but not overlapped) and because the routine must be able to find the proper ring to which to attach a given substituent. When the algorithm is extended to handle general cyclic notations all ring systems, including benzene rings, must be processed with the same push-down stacks. Similarly, all branch symbols must be processed recursively. As each layer of a nested sequence of multiplied strings is encountered the relevant information is pushed onto a stack. The expansion of the notation (i.e., duplication of portions of the connection table) then begins with the innermost layer of the nest, and each recursion expands everything from a given layer in. For example, in notation VI first the portion "/NCU1/\_4" is expanded; then, at the same level, "/SWONUM 3"; next, the entire portion from the first "/" to the second "/"; finally between the triple slashes.\*

WISCO recognizes and processes the following types of multiplicative contraction, an example of each of which is indicated:

1. Multiplication of initial strings (notation VII);
2. Multiplication of terminal strings and side strings (notations VIII and IX);
3. Multiplication on a central asymmetric unit (notations X and XI);
4. Polymeric multiplication (notation XII);
5. Multiplication of strings cited prior to the citation of the ring to which they are attached (notation XIII);
6. Multiplication of strings cited after the citation of the ring to which they are attached (notation XIV);
7. Multiplication of strings attached to rings but with implicit locants (notations XV and XVI).

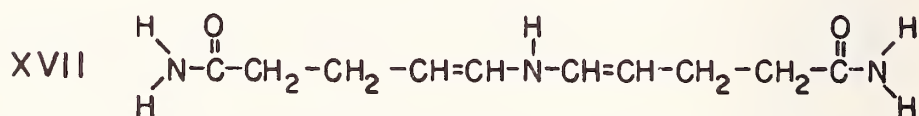
Push-down stacks are also used for processing:

8. Rings--the correspondence between atom number and locant on each ring is stored together with the number of possible substituents at each locant;
9. Branched atom symbols--the atom number of that atom is stored once for each branch attached; and

---

\*The structures to which this and other notations refer are shown in the appendix.

- VI ZVXQ&M1Y&VOYUS&YGUIX///1UNY//O/NCU1/\_4  
X/SWONUM\_3//\_2 ///\_3
- VII ZYUS&O\_2N1O1U1M\_2Y3UYZSONNUM&X
- VIII ZYSHYVQ2M1U1Y/ SO1XZ&NW\_2
- IX Z5X/ MNUNNW\_26YQ
- X QVYQ1U1\_2/PQO/
- XI Z1U1O\_3/YSWM/
- XII ZSW/2VM/\_4SWQ
- XIII WN\_4\_B\_C\_DR\_EZ
- XIV ZOVR\_B-\_C-\_E-\_F-/ VOQ\_4
- XV WN\_5- R\_FZ
- XVI WNR- / Z\_5





10. Variable valence branching symbols--the atom number is stored for each variable valence atom; a flag is stored for each ring. Each atom with variable valence is pushed onto the branched atom stack four times, which exceeds the number of branches permitted by the present algorithm. Whenever the routine has finished with the top entry in the variable valence stack, it erases the record of the corresponding atom from the top of the branched atom stack or of the corresponding ring from the top of the ring stack.

Not all of the above features were successfully implemented. The subroutine which processed the locants used in multiplication type 4 above did not give the proper results; it was removed and has not yet been rewritten. A single atom number for the last atom in a string of type 1 or the first atom in a string of type 2 above, is stored for reference in determining the limits for multiplication. This provision should be modified also to allow for recursion because, e.g., in a nested sequence of type 1 multiplication only the atom number needed for the last-encountered multiplier is now retained.

## 2b. Acyclic Notations

With these conventions in mind, we turn now the processing of a cipher. Consider structure XVII, which is coded as ZV3U1\_2M. The fact that the first character is a Z calls a routine which stores the  $\text{NH}_2$  group as follows:<sup>1</sup>

1	H	( $\emptyset$ ,2)
2	N	( $\emptyset$ ,1) ( $\emptyset$ ,3)
3	H	( $\emptyset$ ,2)

where for each (i,j) couple i represents the bond type for the connection between the current atom and atom j. A single bond is indicated by  $i = \emptyset$ ; double, triple, and nonlocalized multiple bonds are represented by  $i = 2, 3$ , and 1, respectively. If the  $\text{NH}_2$  group had not begun the notation, it would be stored in the order  $\bar{\text{N}}, \text{H}, \text{H}$ , thus terminating a path. But in this case the path must continue from Z with the next atom, and a back connection of 2 (the atom number of the N) is stored. In a similar way, the symbol V is translated as C=O and the connection table becomes:

1	H	( $\emptyset$ ,2)
2	N	( $\emptyset$ ,1) ( $\emptyset$ ,3) ( $\emptyset$ ,4)
3	H	( $\emptyset$ ,2)
4	C	( $\emptyset$ ,2) (2,5)
5	O	(2,4)

---

<sup>1</sup>Actually atomic numbers are stored, not atomic symbols. The numbers of the atoms on the left are for ease of reading only; they do not actually appear in the machine representation.

The symbol 3 calls a subroutine which stores in sequential order as many carbons as indicated by an alkyl numeral. The symbol U causes a bond indicator to be set equal to 2 and processing continues. As each atom is encountered in the symbol string, the back connection is stored in its row of the connection table and its number is stored in the row of the atom so connected to it. Thus the bond indicator is set before the bond to which it refers has been stored. When atom 9 has been reached, the connection table appears thus:

1	H	(0,2)		
2	N	(0,1)	(0,3)	(0,4)
3	H	(0,2)		
4	C	(0,2)	(2,5)	(0,6)
5	O	(2,4)		
6	C	(0,4)	(0,7)	
7	C	(0,6)	(0,8)	
8	C	(0,7)	(2,9)	
9	C	(2,8)		

Next the routine encounters "2", a space followed by a number, which must be a multiplier. Several possibilities are consequently checked for immediately following:

- i) a space followed by a letter, i.e., a locant
- ii) "-R", i.e., implicit locants
- iii) "/", i.e., possible central asymmetric unit
- iv) two spaces, i.e., the end of the cipher.

Any one of these situations would cause specific action. The present example, however, requires further analysis, since the local context of the multiplier admits either a multiplicative contraction of type 1 above or the multiplied side chains of type 2 above. The routine as written pushes the multiplier (2) onto the top of DECML and the most recent atom number (9) onto JKEEP.

A modification of the routine would permit resolution of this situation, since if type 2 multiplicative contraction were the case a slash would have been noted on the slash stack. A slash between the beginning of the notation and the multiplier must be paired with the latter, for the slash delimits the range of the multiplier. If the slash stack is empty the multiplier covers everything back to the beginning of the cipher.

The next character examined is 'M', and the connection table looks as follows after it has been processed:

```

      9  C  (2,8)  (Ø,1Ø)
     1Ø  N  (Ø,9)  (Ø,11)
     11  H  (Ø,1Ø)

```

Now we have reached the end of the cipher. At this point a check of the various stacks is made to see whether there is any outstanding business, in particular whether any multiplication needs to be effected or whether any methyl contractions need to be expanded. If the branched atom stack is not empty, the atom indicated has an implied methyl attached. Once the branched atom stack is empty (as it is immediately in this case), JKEEP is tested. If JKEEP = Ø, ring locant storage is checked for ring multiplication. In our example, JKEEP ≠ Ø, which means that multiplication of type 1 has been defined. (WISCO as presently implemented will not detect type 2 multiplication since it assumes at this point that type 1 is specified. This defect could be eliminated as indicated previously.) A subroutine then copies the portion of the connection table indicated, in our case from atom 1 to atom 9, indicated by JKEEP. A switch indicates that the sequence is to be copied in the reverse direction, i.e., from the end to the beginning. The values of j in the (i,j) couples must be suitably modified during copying. (This routine will work satisfactorily with a slight modification for "forward" copying, but a bug as yet uncorrected prevents proper "reverse" copying.) The end result of the algorithm will be:

```

      1  H  (Ø,2)
      2  N  (Ø,1)  (Ø,3)  (Ø,4)
      3  H  (Ø,2)
      4  C  (Ø,2)  (2,5)  (Ø,6)
      5  O  (2,4)
      6  C  (Ø,4)  (Ø,7)
      7  C  (Ø,6)  (Ø,8)
      8  C  (Ø,7)  (2,9)
      9  C  (2,8)  (Ø,1Ø)
     1Ø  N  (Ø,9)  (Ø,11)  (Ø,12)
     11  H  (Ø,1Ø)
     12  C  (Ø,1Ø)  (2,13)
     13  C  (2,12)  (Ø,14)
     14  C  (Ø,13)  (Ø,15)
     15  C  (Ø,14)  (Ø,16)
     16  C  (Ø,15)  (2,17)  (Ø,19)
     17  O  (2,16)
     18  H  (Ø,19)
     19  N  (Ø,16)  (Ø,18)  (Ø,2Ø)
    2Ø  H  (Ø,18)

```

Type 2 multiplications are handled in an analogous way, with the copy switch set for forward copying.

To see how the routine handles type 3 multiplication, we can generalize from the foregoing example. This type is recognized by a multiplier followed by a slash when the slash stack is empty. The atom number for the last atom symbolized before the multiplier is stored in JKEEP. When the terminal slash is encountered, the last non-terminal atom within the slashes is under current consideration and is joined to a duplicate of the last atom of the repeated string. The copying routine is called with the "reverse" switch set. If the multiplier is greater than 2, then the branched atom stack will indicate the additional points of attachment within the asymmetric central unit (Cf. structure XI). Any residual connections within the slash marks then imply methyl groups.

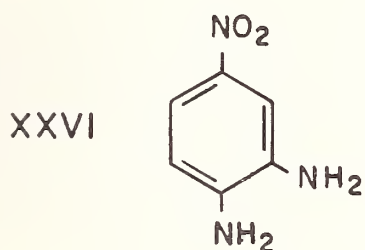
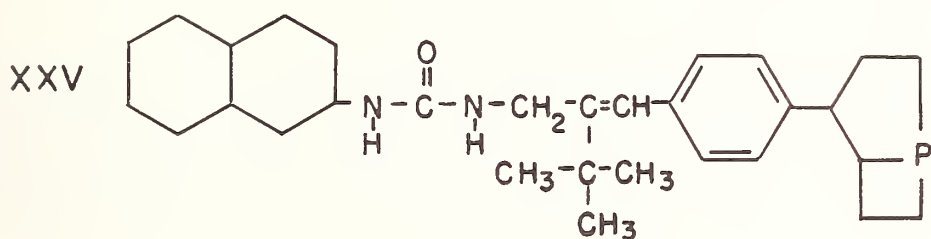
Polymer-type multiplication, type 4, is handled straight-forwardly by "forward" copying of all of the atoms symbolized between the pair of slashes.

Since there is at least a superficial reason to expect confusion between the "&" for an explicitly cited methyl group and the "&" ending a branch containing either a variable valence atom or a ring, we shall consider the relevant contexts. When an "&" (which is not immediately preceded by another "&") is reached there are three possibilities--the preceding atom symbol may be "X", "Y", "K", or "N"; it may be strictly terminal; or it may be a non-terminal symbol other than "X", "Y", "K", or "N". If the preceding atom symbol is one of these branching symbols, then the "&" signals methyl contraction, as do any succeeding ampersands until the branching requirements of the atom have been satisfied. (Cf. notation XVIII.) If the atom symbol preceding the "&" is a non-terminal symbol other than one of these branching symbols, then the first "&" signals the end of a chain. If there is a branching carbon or nitrogen at the top of the branched atom stack, and further "&"s follow, then the succeeding ampersands signal methyl contractions until the branching requirements of the indicated atom have been satisfied. (Cf. notation XIX). For either case of a non-terminal atom symbol preceding the first "&", each further "&" then indicates that all branches attached to one variable valence atom have been cited. (Cf. notations XX and XXI). For each such succeeding "&" the top of the variable valence stack is fetched; if it is a ring flag, then the corresponding ring is discarded from further consideration; if it is an atom number, then the top of the branch symbol stack is discarded as long as this equals the top of the variable valence stack; then the top of the variable valence stack is discarded.

If the atom symbol preceding the first "&" is strictly terminal, then this "&" does not end the chain and it must therefore either represent a methyl group or signal the end of a branch containing either a



XVIII	ZX&&1-SI-QQQ
XIX	ZX2&&-SI-QQQ
XX	Z1YPGY&&&YG-SI-QQQ
XXI	Z1YPGY2&&&YG-SI-QQQ
XXII	QYGV <sub>R</sub> _ER_DG&_CY
XXIII	ZY&2Y1-SE- *2Q2Q2Q&Y&Y1Z
XXIV	ZY&2Y1-SE- W2Q&Y&Y1Z



variable valence atom or a ring (Cf. notation XXII). Note that a methyl contraction must refer to the branch symbol immediately preceding it and that the implied methyl contraction is limited to the last branch symbol (and to "X", "Y", "N", or "K") so that there can be no ambiguity in the meaning of the top of the branched symbol stack.

Rule 8b [2b] appears to require one "&" for each branch point encountered while backing down the branch to the point where ciphering resumes. This complicates the algorithm and appears to be unnecessary to the notation system. It is agreed that one "&" per variable valence branch point would suffice [5b]. However, this modification has not (yet) been adopted.

There is possible ambiguity due to symbols which may or may not be branching symbols depending on valence. We have adopted the asterisk as a flag to follow such a symbol when it is a branching symbol (Cf. notation XXIII), except when the only branches attached are represented by "W" (Cf. notation XXIV).

The existing algorithm has a known bug in its handling of branches. It will decode the notation T45PTJ\_ER\_D1UYX1MVM\_CL66TJ, for structure XXV, in such a way as to yield the connections Y-X-1-M, i.e., it does not in this instance take cognizance of Rule 10 [2b], which demands deletion of the "&"s in a methyl contraction on the only or last branch symbol in the (uncontracted) notation.

In order to remove this bug, a back connection to a branch point must not be made until the notation has been completely processed, i.e., until the end of the cipher has been reached. From the above example it is evident that the last branch point in the notation can be determined only after all characters have been examined. This suggests to us that rules other than Rule 10 may lead to similar situations, and that similar bugs remain to be discovered.

## 2c. Benzene Notations

We begin by considering a simple example: Structure XXVI is ciphered WNR\_CZ\_DZ.

The first two characters are processed as indicated in the preceding section, a subroutine handling "WN" in a manner similar to the handling of an initial "Z". When the "R" is encountered, six carbons are entered into the connection table with the first and last atoms connected to each other and with nonlocalized multiple bonds indicated:

1	O	(2,2)		
2	N	(2,1)	(2,3)	( $\emptyset$ ,4)
3	O	(2,2)		
4	C	( $\emptyset$ ,2)	(1,5)	(1,9)
5	C	(1,4)	(1,6)	
6	C	(1,5)	(1,7)	
7	C	(1,6)	(1,8)	
8	C	(1,7)	(1,9)	
9	C	(1,8)	(1,4)	

Next a temporary block of six words is pushed onto the ring stack. For our example this block would look as follows:

1	$\emptyset$	4	A
$\emptyset$	1	5	B
$\emptyset$	1	6	C
$\emptyset$	1	7	D
$\emptyset$	1	8	E
$\emptyset$	1	9	F

The 1 in the left-hand position of the first word indicates a benzene ring, in anticipation of the routine enlarged for processing general cyclic structures. The next field contains the number of substitutable positions left for each ring atom, decremented each time a substituent is attached. The third field has the atom number corresponding to the locant given in the right-most field. When, in our example, the locant "C" is encountered, the storage block for the current ring is searched for the locant and the corresponding atom number (here 6) is stored as the back connection, the second field is decremented, and processing continues with the substituent symbol following the locant. After the first such substituent has been processed, the connection table appears in part as:

6	C	(1,5)	(1,7)	( $\emptyset$ ,1 $\emptyset$ )
.				
.				
.				
9	C	(1,8)	(1,4)	
1 $\emptyset$	N	( $\emptyset$ ,6)	( $\emptyset$ ,11)	( $\emptyset$ ,12)
11	H	( $\emptyset$ ,1 $\emptyset$ )		
12	H	( $\emptyset$ ,1 $\emptyset$ )		

When an "&" is encountered that finishes a branch containing a ring (Cf. cipher XXII), the current ring block is discarded and processing returns to the previous ring. In this example a block is set up for the first "R" and the locant "E" is interpreted with its aid; then a second block is pushed onto the stack for the second "R". The locant

"\_D" is interpreted from the second block. The "&" causes the second block to be erased, and the locant "\_C" is interpreted again from the first block.

For a cipher with type 5 multiplication (Cf. notation XIII), the locants cited before the ring are stored in a table. When the ring block has been created for the "R", these locants provide the back connections for the repeated string. Type 6 multiplication (Cf. notation XIV) is handled in much the same way except that it is the hyphenated locants that are packed into a table.

The analysis of type 7 multiplication is not debugged in the current version of WISCO. For either of the two cases, expansion of the multiplied group (as intended) must be the last item in processing the notation, because the counts are implicit. When either "-R" or "-/" is encountered, an appropriate flag corresponding to the proper ring can be pushed onto the top of a stack. When the end of the cipher is reached (and all other matters have been attended to), the residual locants on the ring, as indicated by 1's in the second field of the atom words of the ring stack, provide the back connections for the multiplied strings.

The copying routine requires knowing the atom number for the back connection and the limits of the connection table to be copied. In "forward" copying, the replicate of the first atom of the repeated portion is back-connected to the atom indicated above; all other atom numbers in the (i,j) couples are edited during copying by adding a constant to correspond to the position in the connection table of this portion after replication. For "backward" copying, the editing of atom numbers requires subtraction from a constant.

## 2d. General Cyclic Notations

The basis devised for an algorithm to effect the complete analysis of cyclic notations is the separation of those portions of the notation which lie between the delimiters L and J or T and J into distinct fields. Each of these fields deal with one of the several types of features which may be cited as parts of the cyclic system. The results of such an analysis would be used as input to an algorithm for creating an atom-by-atom listing of each cited ring, such as is shown for the special case of linearly fused rings (Cf. below), and for identifying the nature of the bond between each pair of atoms.

The fields which may be present are initial character, ring declaration, shared bridge locants, unshared bridge locants, multicyclic points, last locant, hetero symbols, unsaturation codes, ring saturation code, and final character. For example, in the notation T666\_B6\_C6\_3ABC\_S\_KMJ for the cyclic system shown as structure III, the fields present are: initial character, ring declaration, multicyclic points, last locant, hetero atom, and final character.



The initial character must be an L or T, which is retained for later error checking. The routine then looks for ring information, filling a table with the ring size and the atom number corresponding to the initial (fusion) locant of each ring, in the order of citation. Thus, for our example the ring declaration would be as follows:

<u>Initial Locant</u>	<u>Ring Size</u>
1	6
1	6
1	6
2	6
3	6

The end of this portion of the routine is signaled by any one of the following:

1. Locant followed by a non-numeric character
2. "/"
3. Space followed by a number
4. "K", "M", "N", "O", "P", "S", "V", "X", "Y", or hyphen followed by a non-numeric character
5. "&" or "T"
6. Final character "J"

If the ring declaration is followed by a locant and a non-numeric character (Cf. notation IV), then a shared bridge atom, a ring segment symbol, a hydrogen saturation code, or an unsaturation code follows. Such locants are converted to the corresponding number and stored in another table. Similarly a slash indicates at least one unshared bridge atom (Cf. notation V). A space followed by a number indicates that multicyclic point atoms are cited, and these are similarly stored in a table. For the above example (structure III) the table of such atoms would appear as follows:

1  
2  
3

As with the ring declaration field, these fields are well-defined and their terminations can be detected immediately. If multicyclic point atoms are present, the last locant pertaining to the cyclic system must also be cited. This is similarly converted to a number and stored for future reference. Any ring segment symbols are then picked up and stored along with the numerical equivalent of their locants; in the example on the previous page

11(= K) M

is the only such entry. Similarly citations of H and U or UU are stored with the (converted) locants.

"K" or one of the other symbols listed under possibility 4 indicates a ring segment symbol with implied locant A in a monocycle. A hyphen followed by a non-numeric character presumably introduces a two-letter atomic symbol.

The "T" and "&" symbols are stored and assigned a number according to the order of citation so that they may be paired with the corresponding rings. The occurrence of "J" signals the end of the cyclic portion of the notation, and control must then revert to the calling routine for processing of the tables here generated and of portions of the notation referring to attachments to the cyclic systems, including interpretation of substituent locants. As in the case of benzene rings (Cf. p. 11 above), a temporary storage block would be created and pushed onto the ring-stack for each ring system encountered. The number of words required equals the numeric value of the last locant occurring in the ring system plus one word per branched locant. The last locant may be stated explicitly in the notation, or else the number of atoms in the ring system may be computed via the Landee-Bowman equation [ref. 2b, p. 108].

A routine has been written in ALGOL-60 (Cf. Figure 20) to decode Wiswesser line notations for linearly fused ring aggregates (i.e., ones with neither multicyclic nor enclosed atoms). It assumes that the notation has been analyzed and tabulated as above. On the basis of this information a two-dimensional array (with its size determined by the number of rings and the maximum ring size) is created which lists each ring and the enumeration of the atoms therein.

If the notation were L\_B666J, then the ring declaration table would be:

<u>Initial</u> <u>Locant</u>	<u>Ring</u> <u>Size</u>
2	6
1	6
1	6

This routine would then fill up the array as follows:

The first ring starts with atom number 2 and has size 6:

2 3 4 5 6 7

The second ring starts with atom number 1 so the array would become

2 3 4 5 6 7

1 2 \_ \_ \_ \_

Since atom number 2 has been used in a previous ring, we must get the last atom in that ring and proceed (to six atoms) thus:

2 3 4 5 6 7

1 2 7 8 9 10

Again the next ring starts with atom number 1, but atom number 2 has already been twice used, so we must get the last atom in the last previous ring cited and continue:

2 3 4 5 6 7

1 2 7 8 9 10

1 10 11 12 13 14

A preliminary syntax for this analysis has been developed. This syntax, as given in Table I, will apparently generate (or, equivalently, accept) that set of correct Wiswesser notations which corresponds to the set of possible non-benzene cyclic structures excluding the so-called ring-of-rings structures, whose notations begin with L- or T- . It can be used to formulate an algorithm either to generate or to accept individual notations. The syntax distinguishes in those portions of the notation beginning with L or T and ending with J the following fields, each of which contains a specified type of information:

Initial character (L or T)

Ring declaration

Fusion locant (may be null)

Ring size

Enclosed ring atoms

Unshared bridge atoms

Shared bridge atoms

Multicyclic points and last locant

Ring segment symbol (required if initial character is T)

Carbon saturation (H)

Unsaturation (U or UU)

Ring saturation (& and/or T)

Final character (J)

The first two fields and the last must be present in any non-benzene cyclic notation. The others may or may not be present, depending entirely on the structure under consideration. The symbol D signaling a ring structure with a donor-acceptor bond [3] is not accounted for by this syntax.

Note, however, that this syntax is too powerful, in the sense that it will generate strings of symbols which are not acceptable Wiswesser notations. For instance, any series of ring declarations can be generated, including invalid ones such as in L\_B6J. Implementation of restrictions such as that which requires either <RSS> (see below) to be <NULL> or else <SYMBOL> to be V, X, or Y whenever <IC> is L will require a considerable increase in the complexity of the rules.

The syntax is presented in the so-called Backus Normal Form [7,8]. The metanotational terms (e.g., <IC> for initial character) are enclosed in pointed brackets, and each such appears exactly once on the left-hand side of a rule (i.e., a string of the form ... ::= ... ). The symbol ::= is read "is replaced by" or "is written as". The vertical line indicates alternatives; concatenation means "and". Thus the rule

$$\langle \text{IC} \rangle ::= \text{L} | \text{T}$$

means that any occurrence of the symbol <IC> is replaced by L or T; the rule

$$\langle \text{UBA} \rangle ::= / \langle \text{ATLOC} \rangle$$

indicates that <UBA> is rewritten as / followed by <ATLOC>.

Note that some rules are recursive: the symbol on the left of a given rule may also appear in the right-hand side thereof, as in

$$\langle \text{HYP} \rangle ::= - | - \langle \text{HYP} \rangle.$$



This rule gives rise to the string of hyphens which indicate a branched locant at a given depth. To apply the rule, rewrite <HYP> as either - or <HYP> . If the former alternative is chosen, the recursion ends; if not, reapply <HYP> . This sequence is continued until a string of hyphens of the desired length is obtained. Finally, for the fields which are optionally present the alternative <NULL> is given, as in the case of <ERA>, to allow for their non-occurrence.

As indicated above, this syntax can be used in an algorithm to generate individual notations. The algorithm used to parse a given string will in effect be the inverse of the rule-application procedure outlined above, so that the notation will be accepted as input and will be parsed (i.e., accepted) if and only if a "path" through the syntax can be found such that the path terminates with the rule

$$\langle \text{WLN} \rangle ::= \langle \text{IC} \rangle \dots \langle \text{TC} \rangle$$

An algorithm to "turn around" a procedure of this kind has been discussed by Schneider [9].

The workings of the syntax will be illustrated by tracing the path which would be taken to generate the string T66\_BMT&J . Starting at <WLN> , one is led first to <IC> ; T is generated as the terminal symbol. The next symbol in the definition of <WLN> is <RD> ; the <LOC> <RSIZE> <RD> option must be generated in order to get the substring of two ring sizes. <LOC> must generate <NULL> to account for the suppressed \_A locant, and <RSIZE> must generate <RNGMAG> to get the single digit 6 as a ring size. Going through <RD> again, generating <LOC> <RSIZE> , and proceeding as above generates the second 6 with a suppressed \_A locant. Since no enclosed ring atoms nor multicyclic points are cited. <ERA> and <MCP> terminate via the <NULL> options in each case. In the case of <RSS> the first option allowing recursion is required in order to cite the locant for the M. The symbol <SPACE> terminates immediately with \_ ; <LOCMAG> generates <LETTER> , which terminates with B for the locant. The symbol <SYMBOL> generates <WISLET> which terminates with M. Upon recursion through <RSS> the <NULL> option is generated. Since there are no H-saturations or explicit unsaturations in the cipher the <NULL> options are generated for <HSAT> and <UNSAT> . <ANTSAT> generates first the string T<ANTSAT> ; upon recursion the string &<ANSAT> is generated. This part of the generation is stopped by generating the <NULL> option for <ANTSAT> . The generation is completed by applying the rule <TC> ::=J. For the tree of this generation, see Figure 29.

Table I. Preliminary Syntax for Analysis of  
Cyclic Portions of Wiswesser Notations

Wiswesser Notation String:

$\langle \text{WLN} \rangle ::= \langle \text{IC} \rangle \langle \text{RD} \rangle \langle \text{ERA} \rangle \langle \text{MCP} \rangle \langle \text{RSS} \rangle \langle \text{HSAT} \rangle \langle \text{UNSAT} \rangle \langle \text{ANTSAT} \rangle \langle \text{TC} \rangle ;$

Initial Character Code:

$\langle \text{IC} \rangle ::= \text{L} | \text{T} ;$

Ring Declaration:

$\langle \text{RD} \rangle ::= \langle \text{LOC} \rangle \langle \text{RSIZE} \rangle | \langle \text{LOC} \rangle \langle \text{RSIZE} \rangle \langle \text{RD} \rangle ;$

Locant:

$\langle \text{LOC} \rangle ::= \langle \text{NULL} \rangle | \langle \text{SPACE} \rangle \langle \text{LOCMAG} \rangle ;$

$\langle \text{NULL} \rangle ::= ;$

$\langle \text{SPACE} \rangle ::= \_ ;$

$\langle \text{LOCMAG} \rangle ::= \langle \text{LETTER} \rangle | \langle \text{LETTER} \rangle \langle \text{MOD} \rangle ;$

$\langle \text{LETTER} \rangle ::= \text{A} | \text{B} | \text{C} | \text{D} | \text{E} | \text{F} | \text{G} | \text{H} | \text{I} | \text{J} | \text{K} | \text{L} | \text{M} | \text{N} | \text{O} | \text{P} | \text{Q} | \text{R} | \text{S} | \text{T} | \text{U} | \text{V} | \text{W} ;$

$\langle \text{MOD} \rangle ::= \langle \text{AMP} \rangle | \langle \text{HYP} \rangle | \langle \text{NULL} \rangle ;$

$\langle \text{AMP} \rangle ::= \& | \& \langle \text{MOD} \rangle ;$

$\langle \text{HYP} \rangle ::= - | - \langle \text{HYP} \rangle ;$

Ring Size:

$\langle \text{RSIZE} \rangle ::= \langle \text{RNGMAG} \rangle | - \langle \text{NONZERO} \rangle \langle \text{DIGIT} \rangle - ;$

$\langle \text{RNGMAG} \rangle ::= 3 | 4 | 5 | 6 | 7 | 8 | 9 ;$

$\langle \text{NONZERO} \rangle ::= 1 | 2 | \langle \text{RNGMAG} \rangle ;$

$\langle \text{DIGIT} \rangle ::= \emptyset | \langle \text{NONZERO} \rangle ;$

Enclosed Ring Atom:

$\langle \text{ERA} \rangle ::= \langle \text{NULL} \rangle | \langle \text{UBA} \rangle | \langle \text{SBA} \rangle | \langle \text{UBA} \rangle \langle \text{SBA} \rangle ;$

Unshared Bridge Atoms:

$\langle \text{UBA} \rangle ::= / \langle \text{ATLOC} \rangle ;$

$\langle \text{ATLOC} \rangle ::= \langle \text{LOCMAG} \rangle | \langle \text{LOCMAG} \rangle \langle \text{ATLOC} \rangle ;$

Table I. (Continued)

Shared Bridge Atoms:

$\langle \text{SBA} \rangle ::= \langle \text{LOCMAG} \rangle | \langle \text{LOCMAG} \rangle \langle \text{SBA} \rangle ;$

Multicyclic Point Atom:

$\langle \text{MCP} \rangle ::= \langle \text{NULL} \rangle | \langle \text{MCP2} \rangle \langle \text{SPACE} \rangle \langle \text{LOCMAG} \rangle ;$

$\langle \text{MCP2} \rangle ::= +1 \langle \text{LOCMAG} \rangle | +1 \langle \text{MCP2} \rangle \langle \text{LOCMAG} \rangle ;$

[Note: n successive occurrences of +1 are written as the number n.]

Ring Segment Symbols:

$\langle \text{RSS} \rangle ::= \langle \text{NULL} \rangle | \langle \text{SPACE} \rangle \langle \text{LOCMAG} \rangle \langle \text{SYMBOL} \rangle \langle \text{RSS} \rangle | \langle \text{SYMBOL} \rangle \langle \text{RSS} \rangle ;$

$\langle \text{SYMBOL} \rangle ::= \langle \text{WISLET} \rangle | - \langle \text{ABC} \rangle \langle \text{ABC} \rangle - ;$

$\langle \text{WISLET} \rangle ::= \text{A} | \text{B} | \text{H} | \text{K} | \text{M} | \text{N} | \text{O} | \text{P} | \text{S} | \text{V} | \text{W} | \text{X} | \text{Y} | \langle \text{HAL} \rangle ;$

$\langle \text{HAL} \rangle ::= \text{E} | \text{F} | \text{G} | \text{I} ;$

$\langle \text{ABC} \rangle ::= \langle \text{LETTER} \rangle | \text{X} | \text{Y} | \text{Z} ;$

Hydrogen Saturation:

$\langle \text{HSAT} \rangle ::= \langle \text{NULL} \rangle | \langle \text{SPACE} \rangle \langle \text{LOCMAG} \rangle \text{H} \langle \text{HSAT} \rangle ;$

Unsaturation Code:

$\langle \text{UNSAT} \rangle ::= \langle \text{NULL} \rangle | \langle \text{SPACE} \rangle \langle \text{LOCMAG} \rangle \langle \text{USYM} \rangle \langle \text{EXLOC} \rangle \langle \text{UNSAT} \rangle ;$

$\langle \text{USYM} \rangle ::= \text{U} | \text{UU} ;$

$\langle \text{EXLOC} \rangle ::= \langle \text{NULL} \rangle | - \langle \text{LOCMAG} \rangle ;$

Ampersand and T unsaturation:

$\langle \text{ANTSAT} \rangle ::= \langle \text{NULL} \rangle | \& \langle \text{ANTSAT} \rangle | \text{T} \langle \text{ANTSAT} \rangle ;$

Terminal Character Code:

$\langle \text{TC} \rangle ::= \text{J}$

# REFERENCES

- 1a Hyde, E., F.W. Matthews, L.H. Thomson and W.J. Wiswesser, "Conversion of Wiswesser Notation to a Connectivity Matrix for Organic Compounds", in Abstracts of Papers, American Chemical Society, 153rd National Meeting, Miami Beach, Fla., April 9-14, 1967, paper F13.
- 1b Landee, F.A., "Computer Programs for Handling Chemical Structures Expressed in the Wiswesser Notation", in Abstracts of Papers, American Chemical Society, 147th National Meeting, Philadelphia, Pa., April 5-10, 1964, paper 3F.
- 2a Wiswesser, W.J., "Line-Formula Chemical Notation," revision of October 1964 by E.G. Smith et al., of Wiswesser's Monograph, title as above, published by Thomas Y. Crowell Co., New York, 1954.
- 2b Wiswesser, W.J., "Line-Formula Chemical Notation," revision of April 1965 by E.G. Smith et al., of Wiswesser's Monograph, title as above, published by Thomas Y. Crowell Co., New York, 1954.
- 3 Landee, F.A., C. Bowman, and M.H. Reslock, Private Communication, November 1965.
- 4 Granito, C., Private Communication.
- 5a Smith, E.G., Private Communication, June 1964.
- 5b Smith, E.G., Private Communication, October 1966.
- 6 Tauber, S.J., G.R. Bolotsky, G.F. Fraction, C.L. Kirby, and G.R. Reed, "Algorithms for Utilizing Hayward Chemical Structure Notations", in H. Pfeffer, ed. "Information Retrieval among Examining Patent Offices", (Proceedings of the Fifth Annual Meeting (1965) of the Committee for International Cooperation in Information Retrieval Among Examining Patent Offices), Thompson Books, Inc., Washington, D.C., 1966, p. 351-394.
- 7 Naur, P., et al., "Report on the Algorithmic Language ALGOL-60", Comm.ACM, 3, 299-314 (May, 1960); esp. p. 301.

- 8 Backus, J.W., "The Syntax and Semantics of the Proposed International Algebraic Language of the Zurich ACM-GAMM Conference", in "Information Processing", (Proceedings of the International Conference 1959), UNESCO, Paris, 1960, p. 125-132; esp. p. 129.
- 9 Schneider, V.B., "Pushdown-Store Processors of Context-Free Languages", Ph.D. Dissertation, Northwestern University, June, 1966.

#### Acknowledgment

The research described in this report was financed in part by the National Science Foundation under Transfers of Funds GN-455 and AG-69.





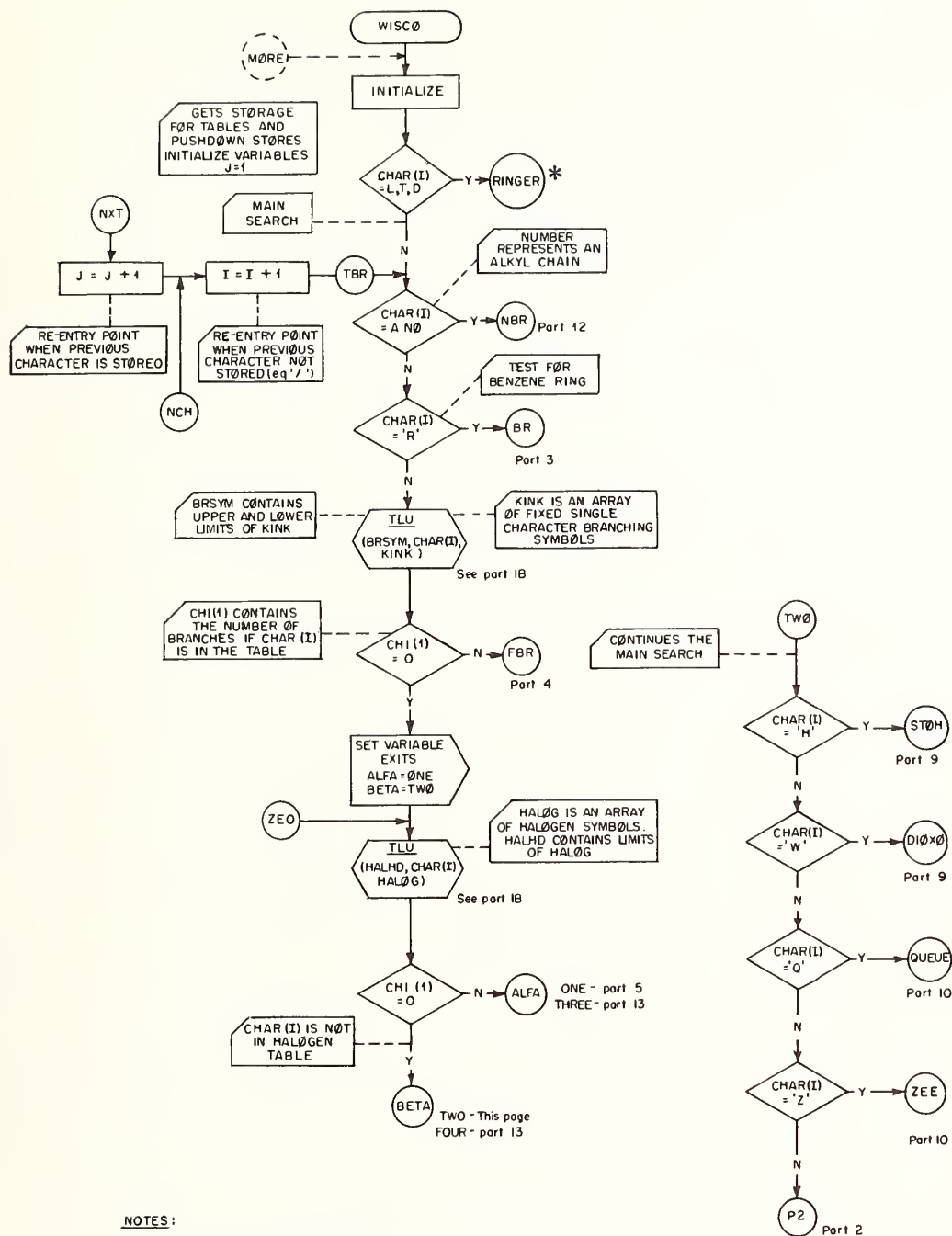


FIGURE 1. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS INTO CONNECTION TABLES. Part I

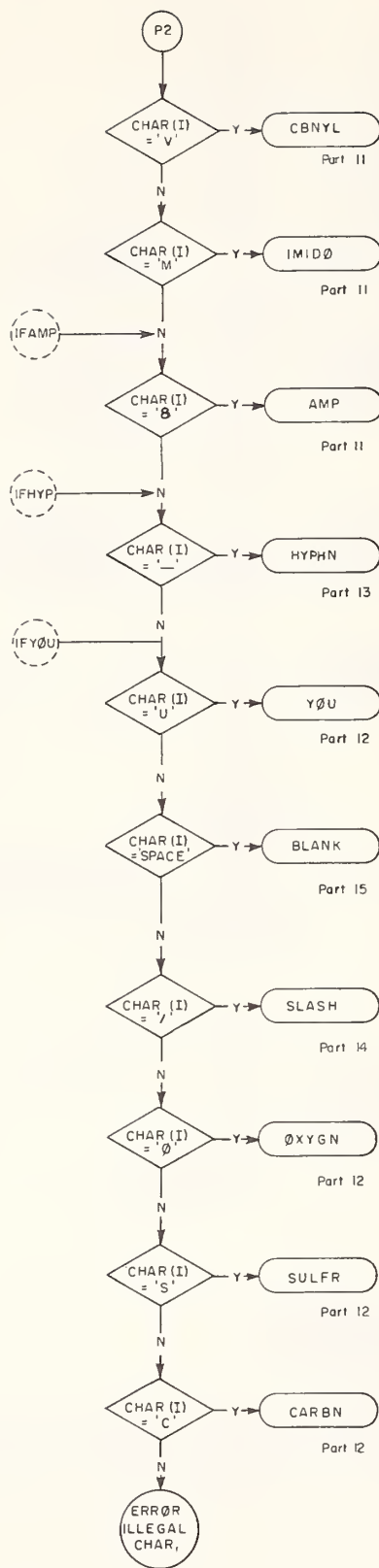


FIGURE 2. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS-Part 2

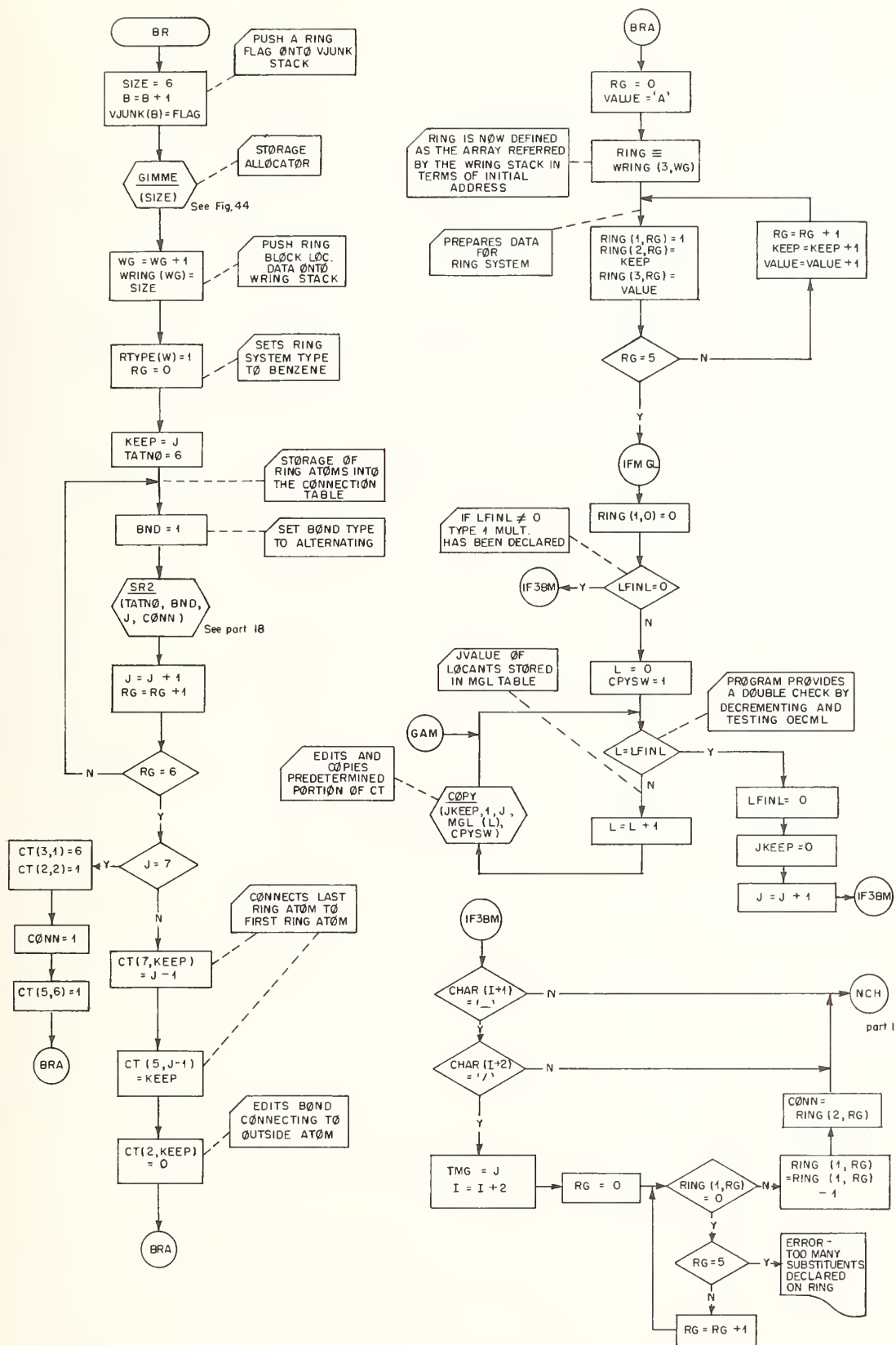


FIGURE 3. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS-Part 3

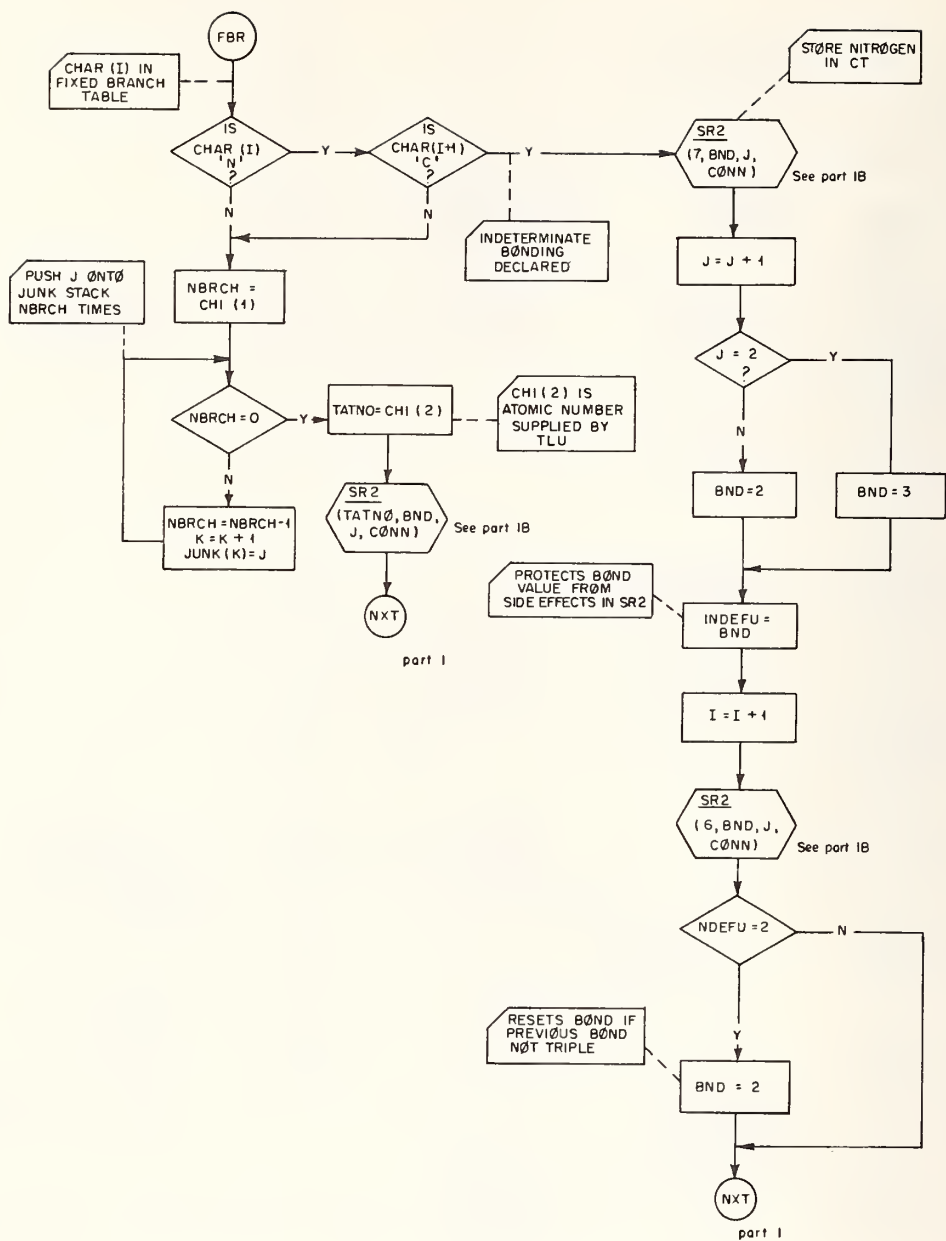


FIGURE 4. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS -Part 4





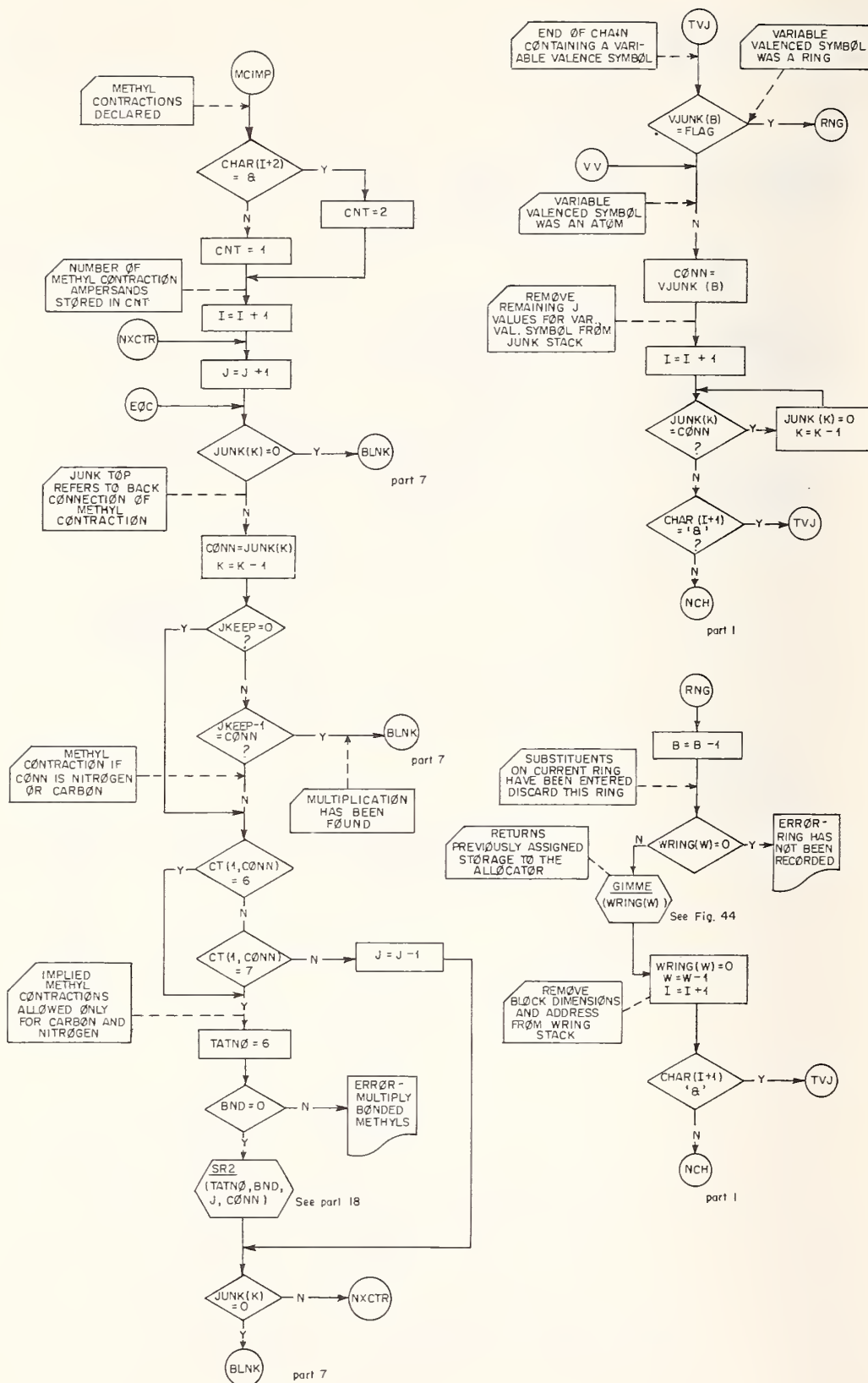


FIGURE 6. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS -Part 6

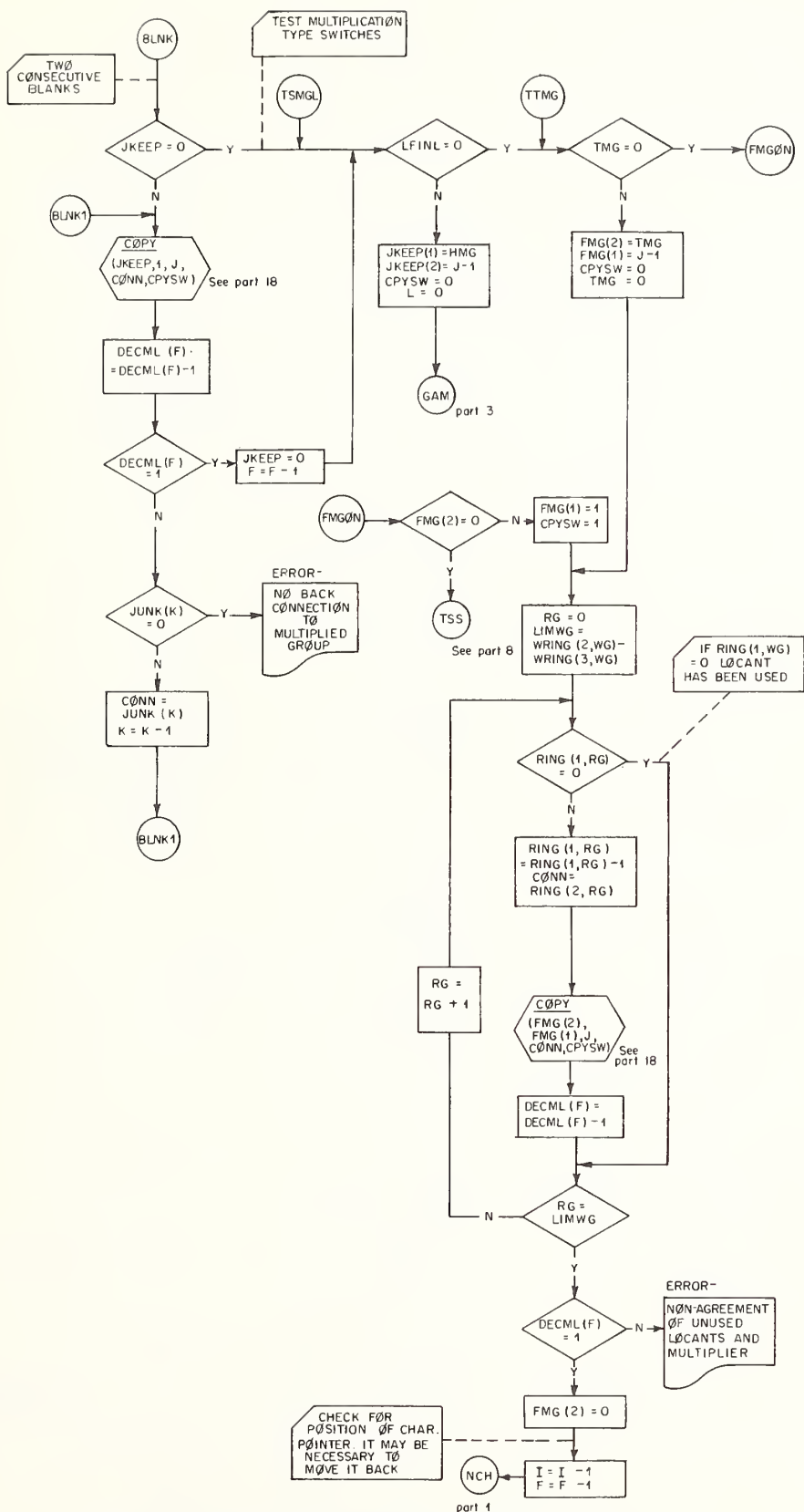


FIGURE 7. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS -Part 7

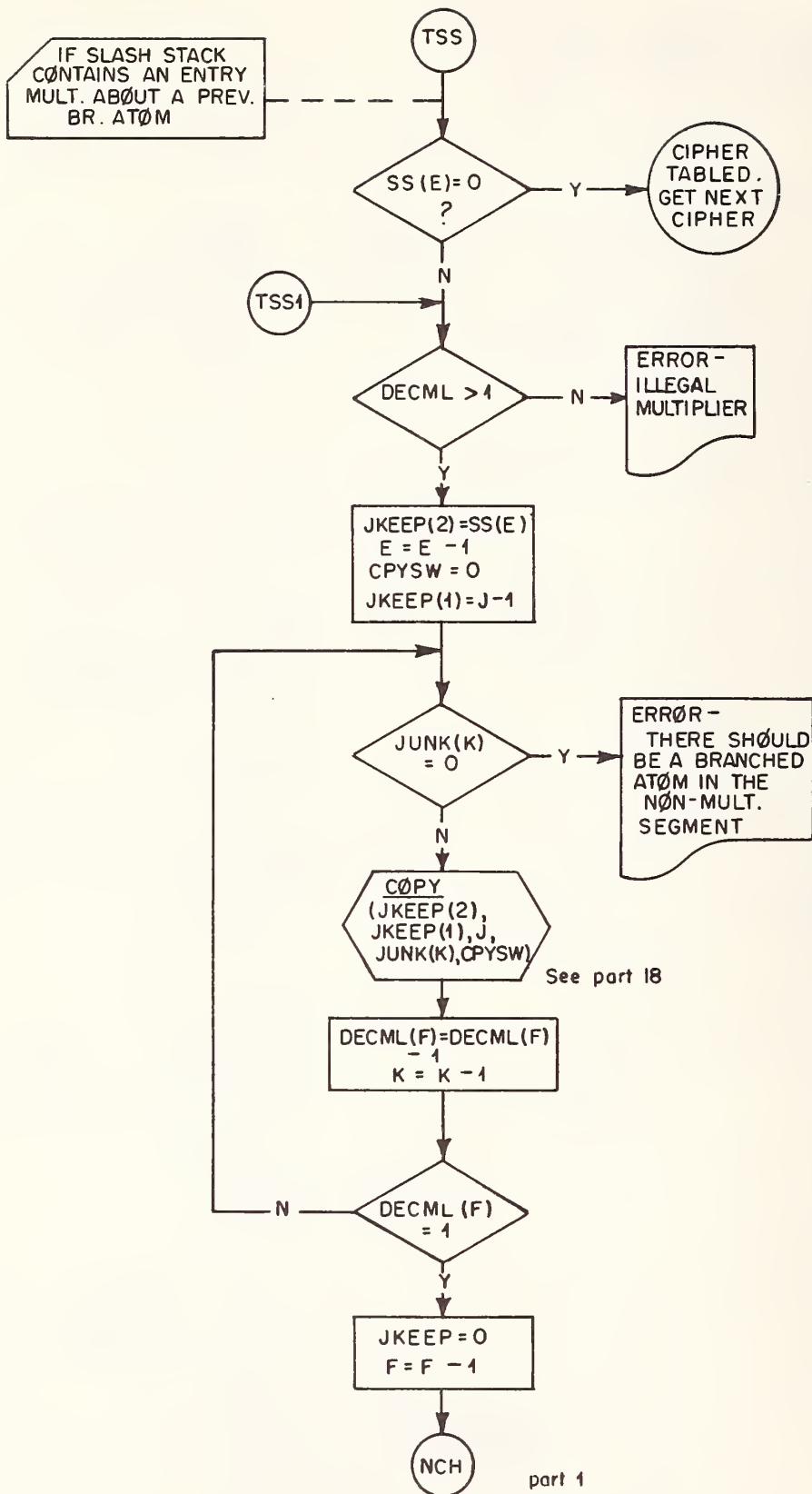


FIGURE 8. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS - Part 8

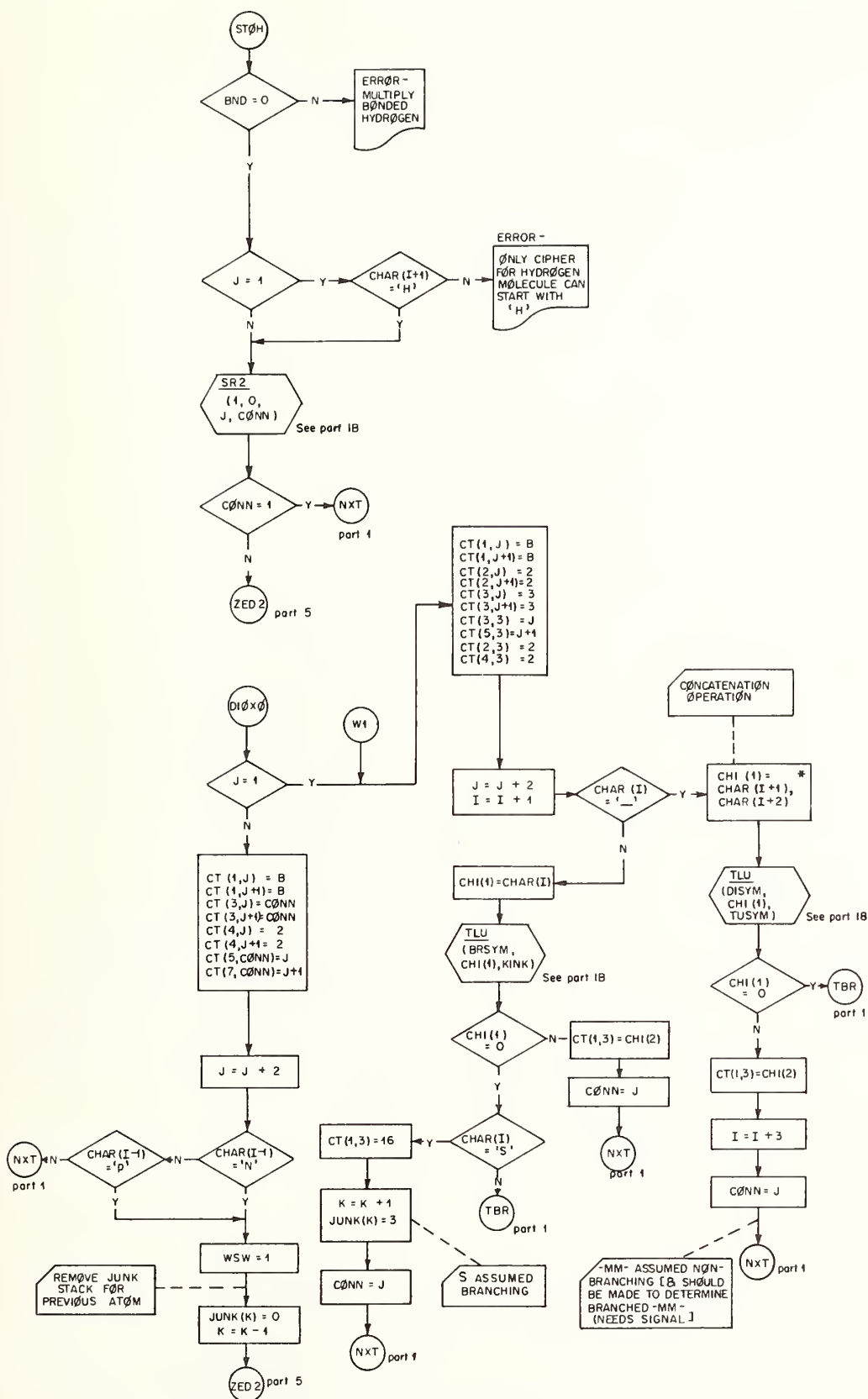


FIGURE 9. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS-Part 9



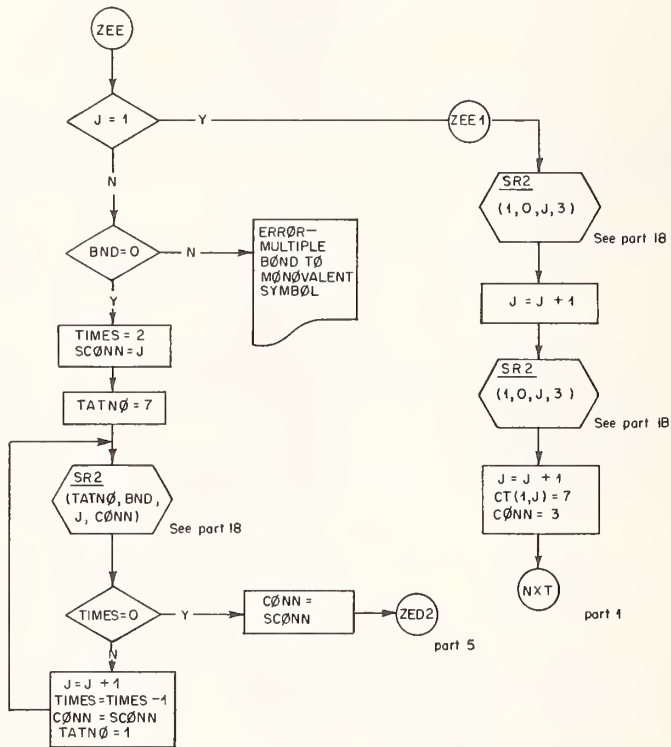
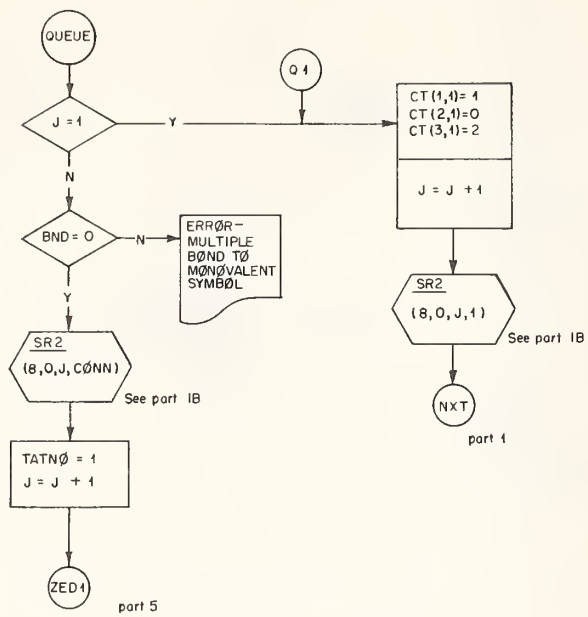


FIGURE 10. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS - Part 10

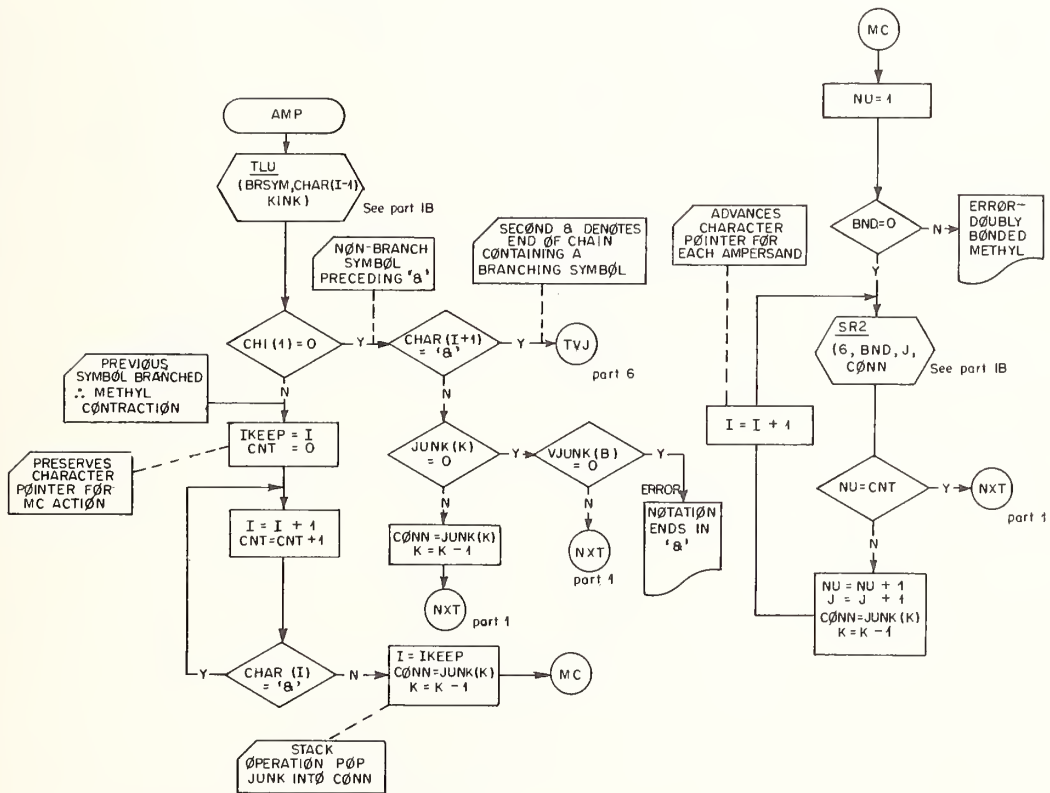
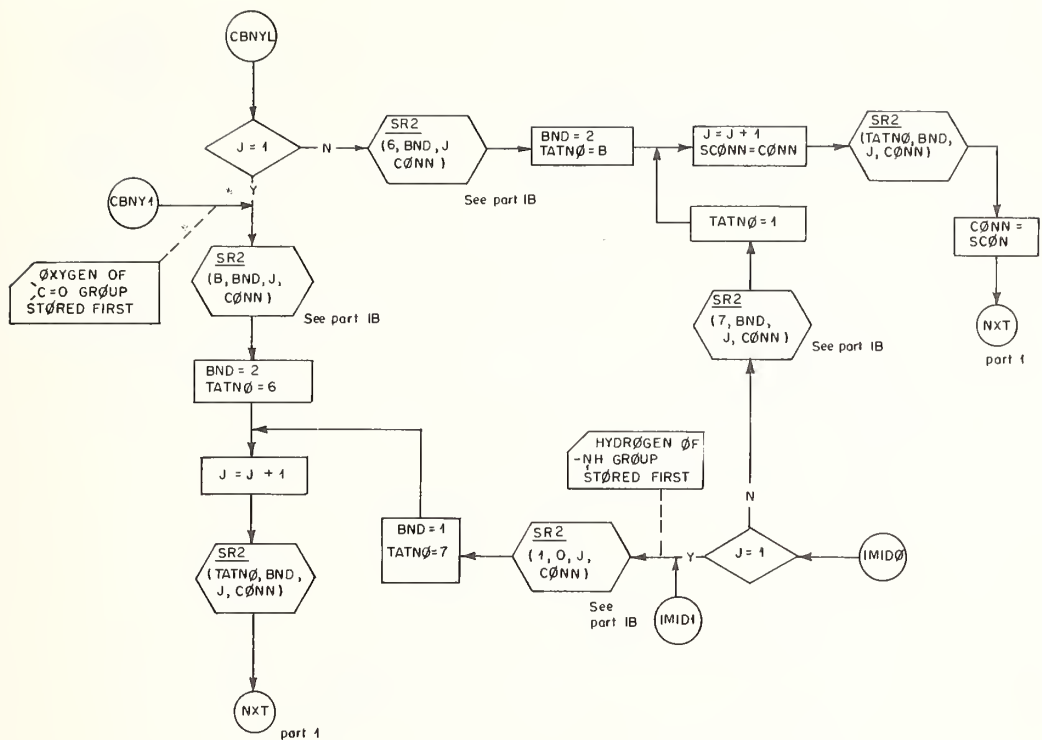


FIGURE 11. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS-Part 11

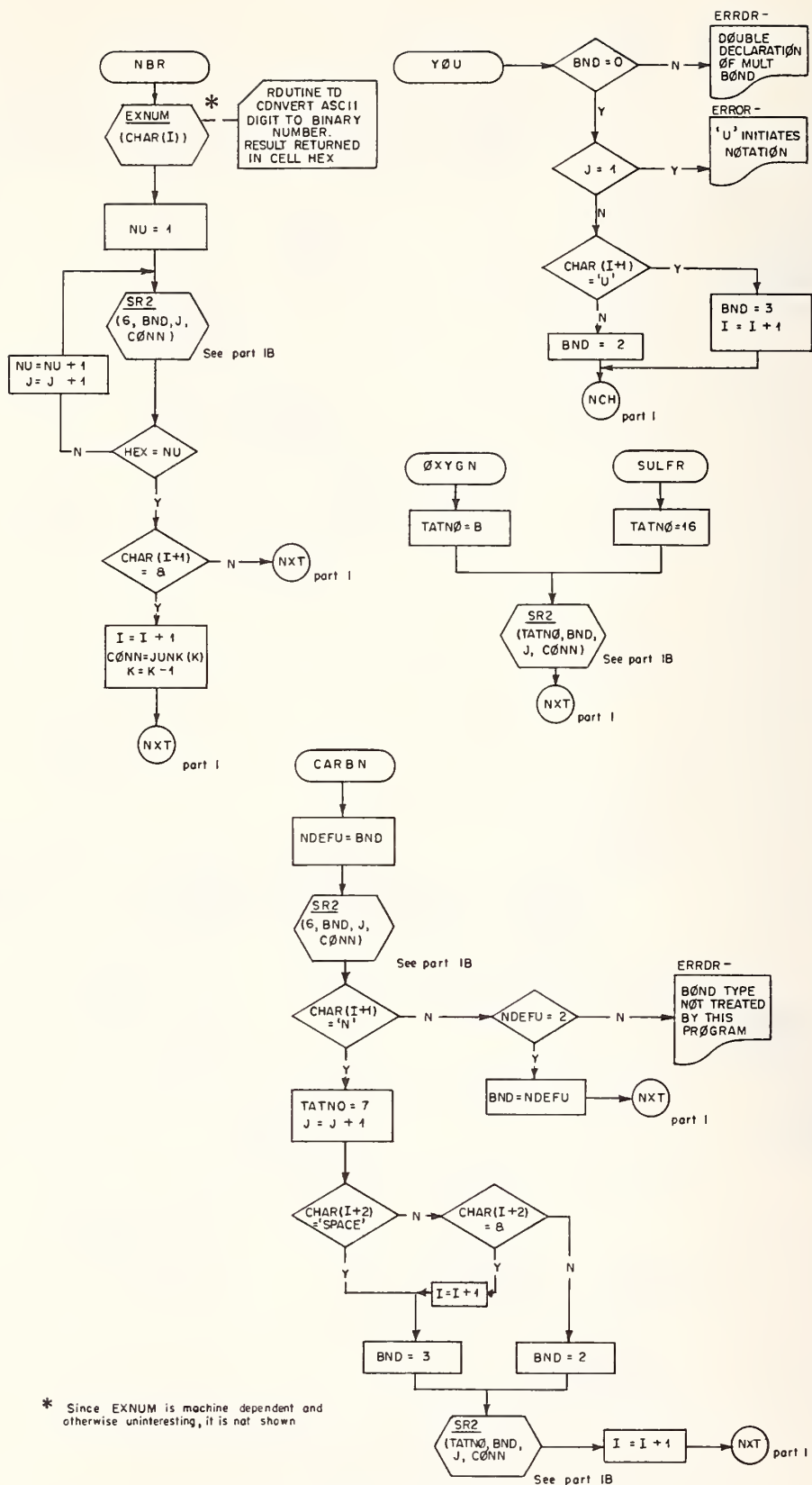


FIGURE 12. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS - Part 12

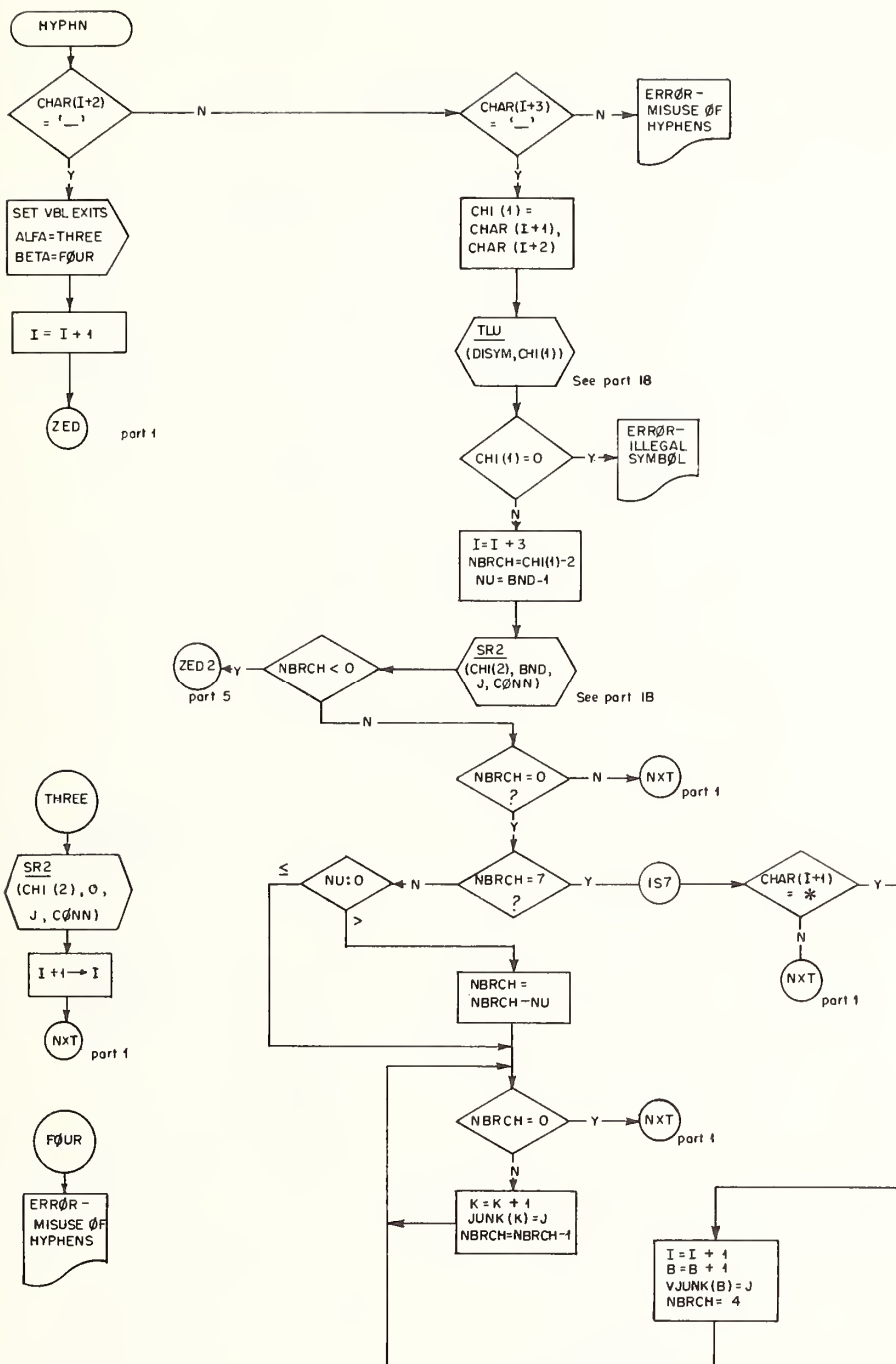


FIGURE 13. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS -Part 13

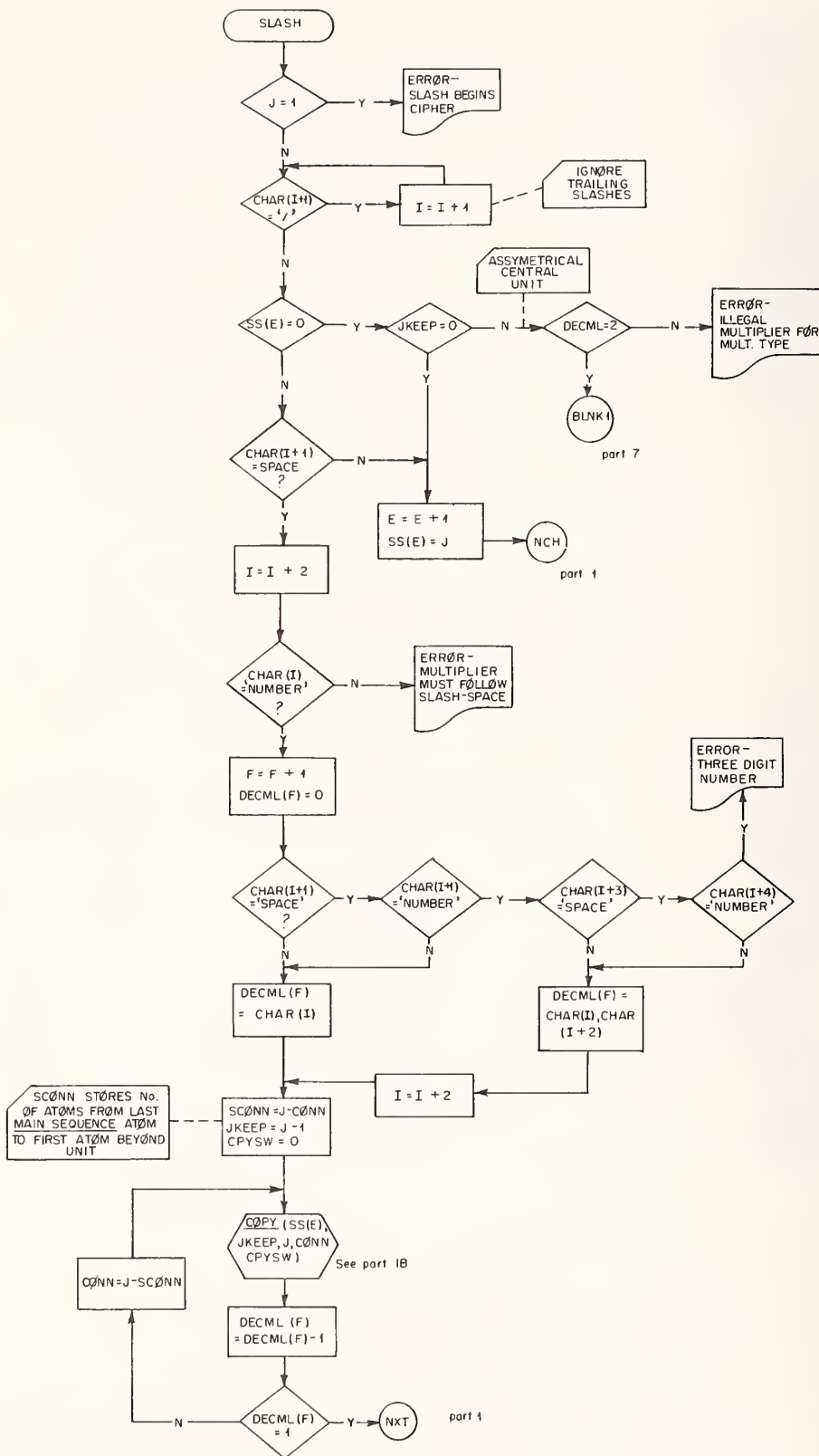


FIGURE 14. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS-Part 14









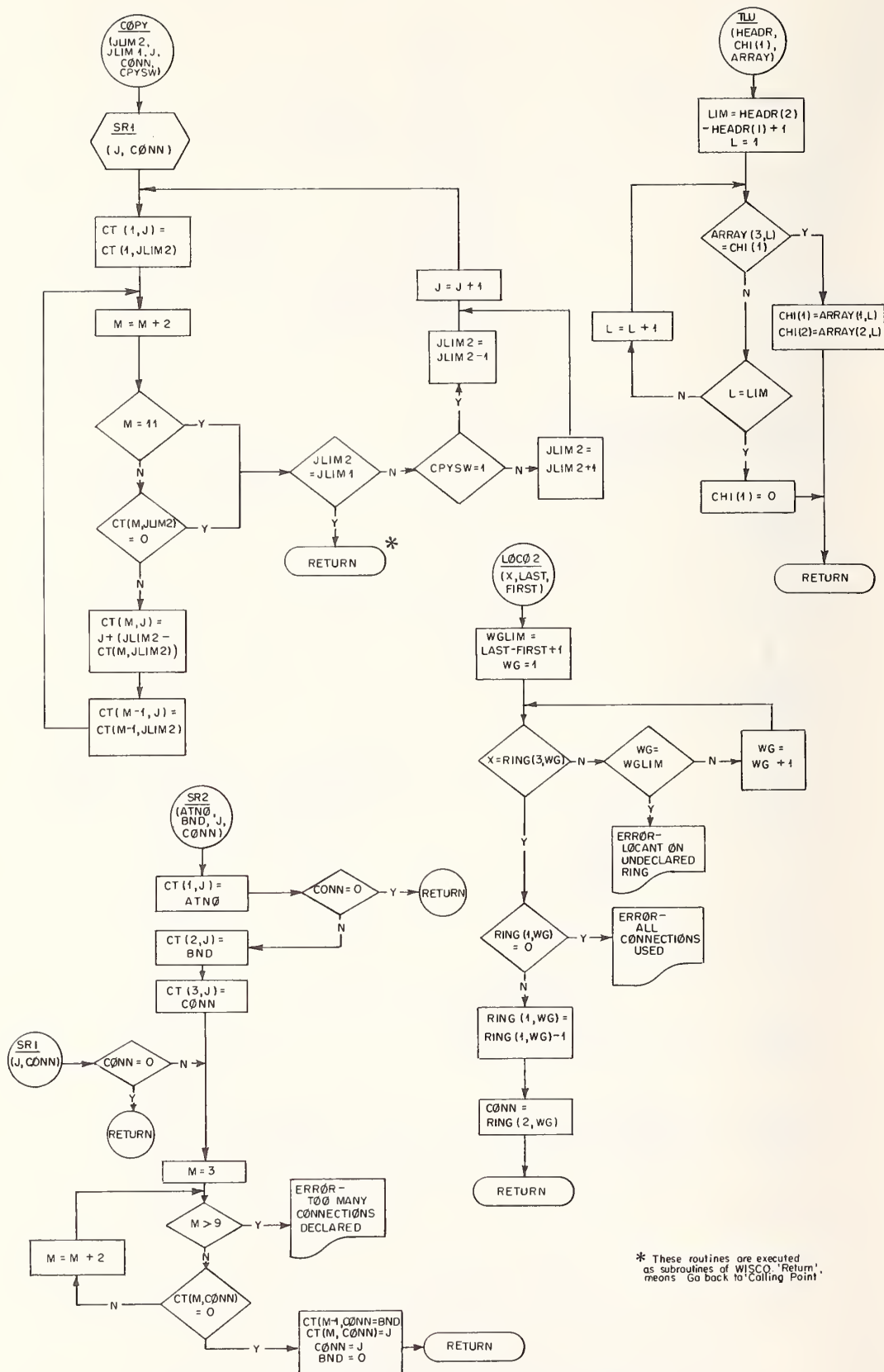


FIGURE 18. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS -Part 18

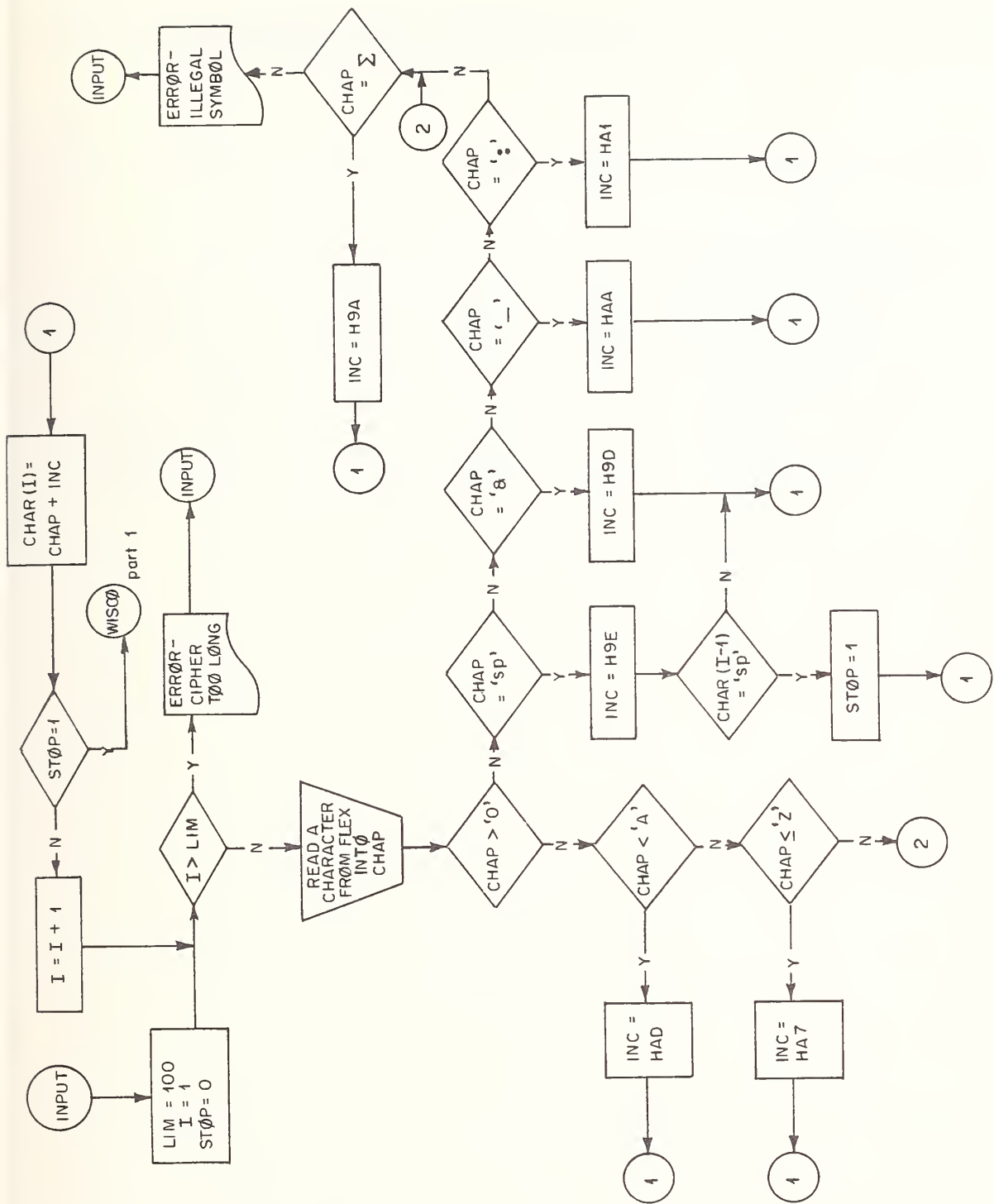


FIGURE 19. FLOW-CHART FOR TRANSFORMING WISWESSER NOTATIONS -Part 19

FIGURE 20. ALGOL-60 PROGRAM FOR CONVERTING WISWESSER LINE NOTATIONS OF LINEARLY FUSED RING SYSTEMS TO CONNECTION TABLES.

Integer-procedure DECODE(SIZE); comment Initial locant: (ILOC).  
Number of rings: (N);

Comment This procedure is for decoding linear strings of fused rings coded in WLN. All ring sizes and the initial locant of each ring are assumed to have been stored, by earlier parts of the general program in SIZE(R) and ILOC(R) respectively, where  $0 < R \leq N$  ( $N$  = the number of rings in the fused ring aggregate). This procedure fills a two-dimensional array RING with the enumeration of each atom in each ring. RING is so ordered that within each row every atom is connected to its neighbor, with the last atom in the row being connected to the first atom in the row, i.e., RING(R,J) is connected to RING(R, $J \oplus 1$ ) and RING(R, $J \ominus 1$ ), where  $\oplus$  and  $\ominus$  define arithmetic operations modulo SIZE(R);

value N; integer USED,FUSED,N,K,R,J;

integer array SIZE,ILOC[1:N],RING[1:N,1:R];

begin

USED:=0;

R:=1;

GAMMA:K:= ILOC [R];

J:=1;

BETA:RING[R,J]:= K;

if K = USED then K:=FUSED else begin

if J = SIZE[R] then begin

if R=N then go to STOP else begin

FUSED:=K;

USED:=ILOC [R];

R:= R+1; go to GAMMA end end else

K:= K+1 end

DELTA: J:= J+1; go to BETA;

STOP: end DECODE



Notes for Figures 21 through 25:  
Flow Chart Subroutine To Examine  
Storage After Processing a Wiswesser Notation

DEBUG - A Subroutine of WISCO

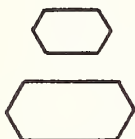
SYMTAB - A table generated at assembly time by the assembler. The length of the table is stored in a location called SYMEND. Each word of the table (SYMTAB(I)) has three fields: SYMTAB(I)<sub>1</sub> contains the name of a location or of a storage area; SYMTAB(I)<sub>2</sub> contains the address corresponding to the name; SYMTAB(I)<sub>3</sub> contains the number of words accessed by the name if a storage area or a flag (all ones) if a named location (i.e., the address of an instruction).



encloses a message to be printed out on-line.



encloses instructions for on-line input.



} encloses operations which are machine-dependent and are indicated only in a general statement.



- - contains comments intended to explain a set of operations and create perspective.

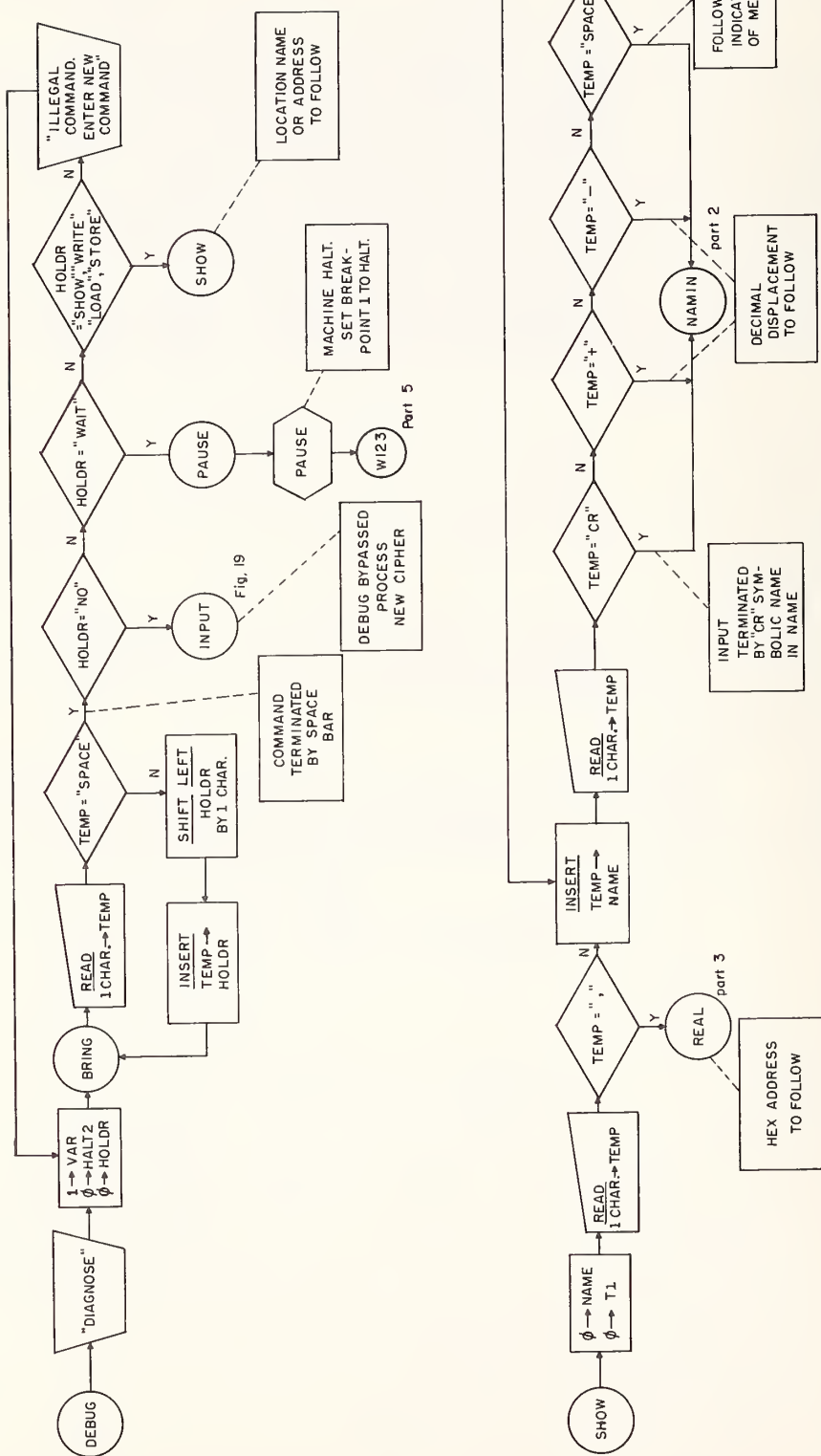


FIGURE 21. FLOW-CHART FOR SUB-ROUTINE TO EXAMINE STORAGE AFTER PROCESSING A WISSESSER NOTATION - Part I

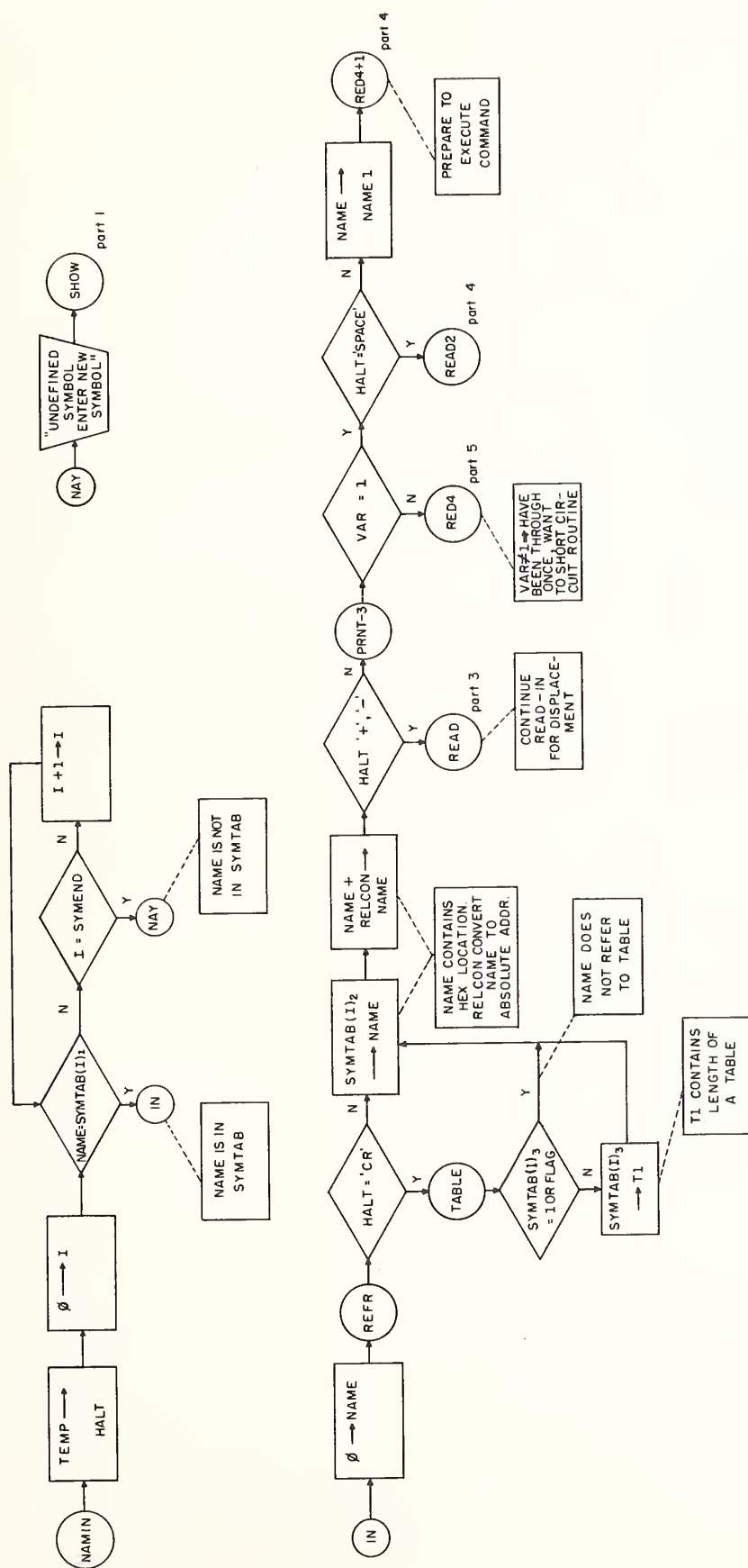


FIGURE 22. FLOW - CHART FOR SUB-ROUTINE TO EXAMINE STORAGE - PART 2



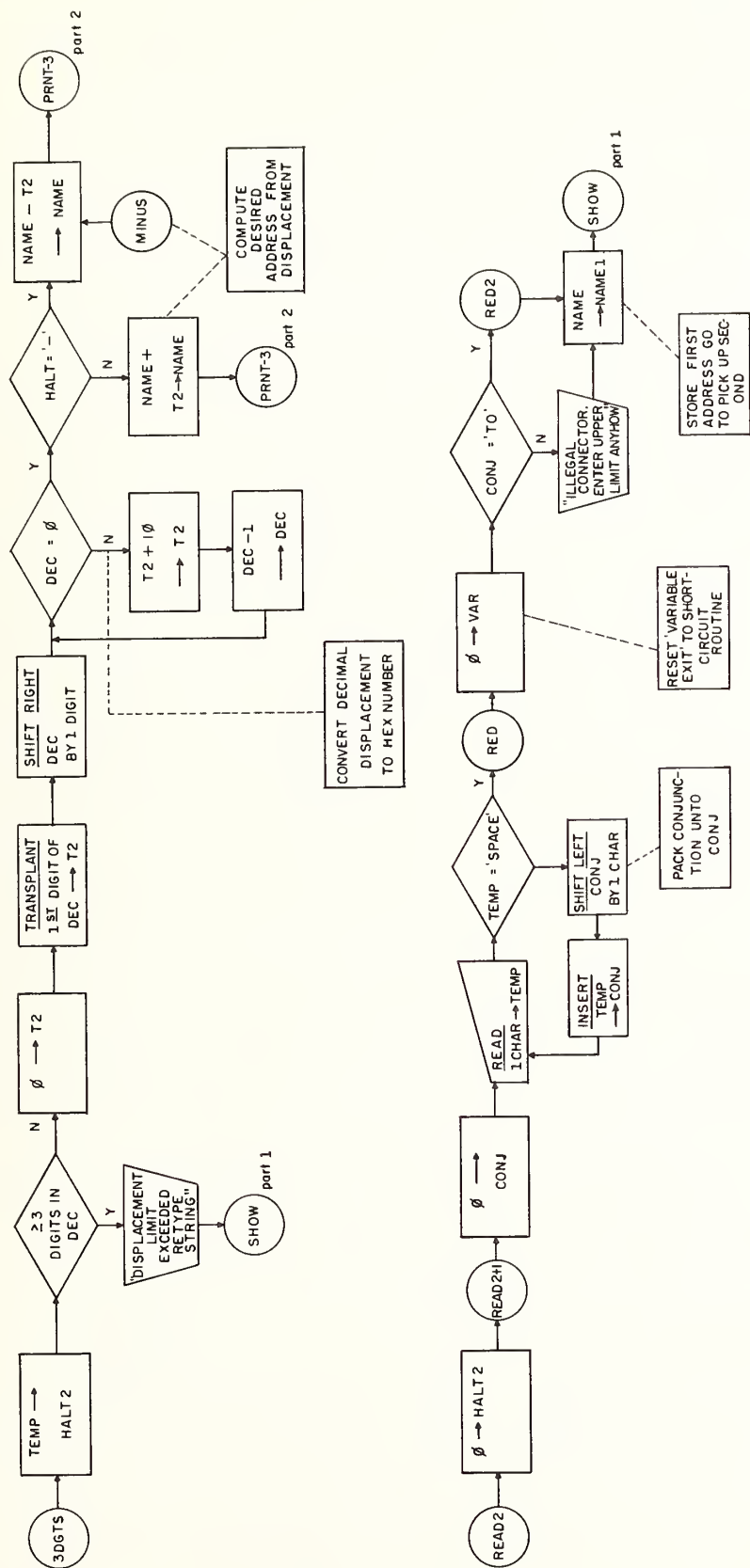


FIGURE 24. FLOW-CHART FOR SUB-ROUTINE TO EXAMINE STORAGE - Part 4

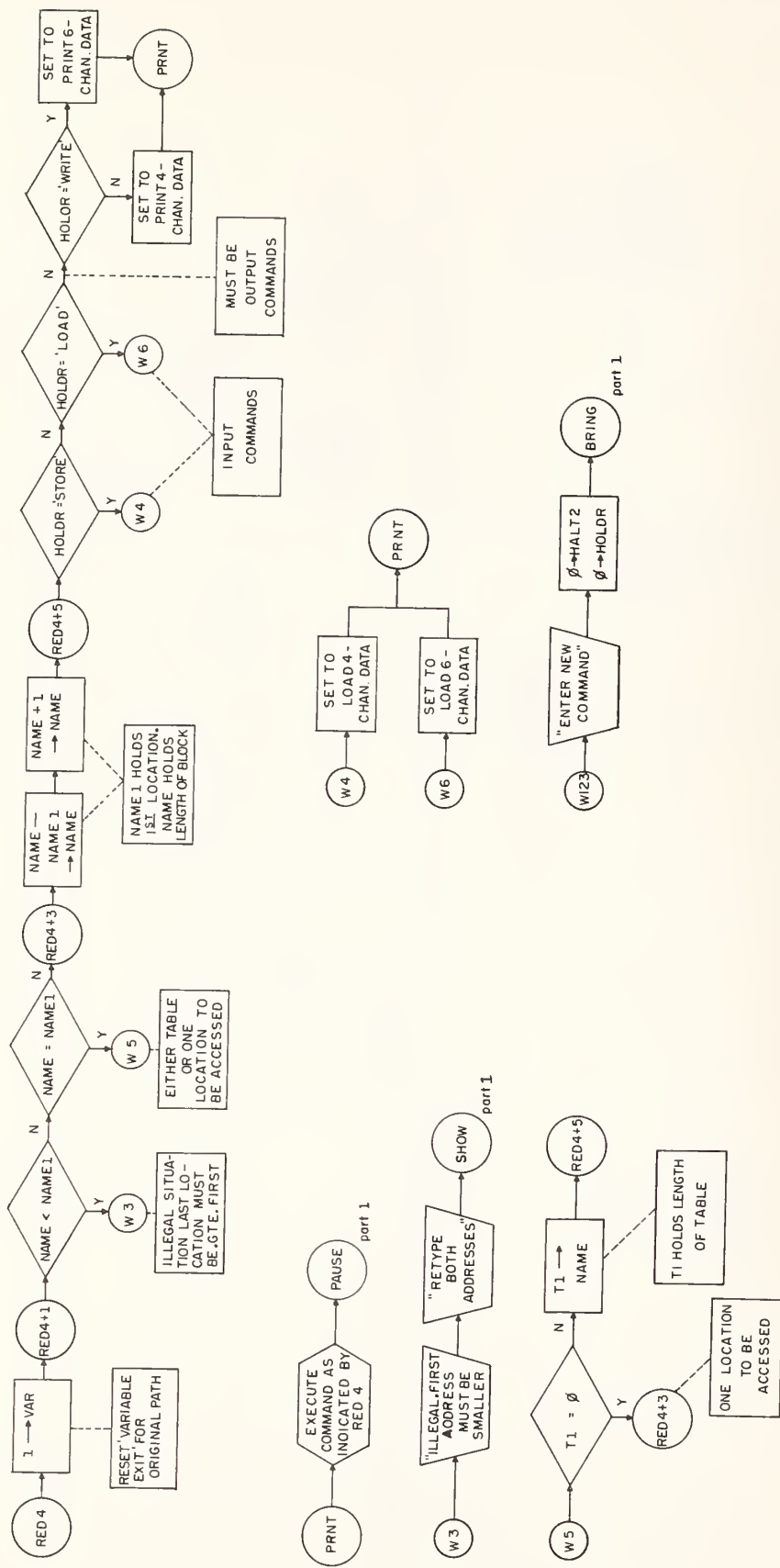


FIGURE 25. FLOW-CHART FOR SUB-ROUTINE TO EXAMINE STORAGE - PART 5

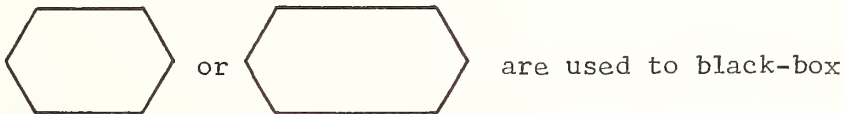


Notes for Figures 26 through 28:  
Flow Chart for Pseudo-Dynamic Storage Allocation

GIMME - A Subroutine of WISCO (Adaptable to a general Storage Allocator)

Common cell SIZE contains either number of words requested or first address of storage block being returned. In the first case, control is returned to calling point with table limits in SIZE. In the second, the existence of the table indicated is deleted from system records.

INUSE is a block of 20 locations used by GIMME to store pointers to the tables it has created. Copies of these pointers are returned via SIZE to the calling routine.



operations which are machine-dependent, knowledge of which being unessential to the understanding of GIMME's operation.

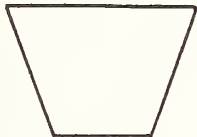
Boxes of the form



enclose comments

Which are intended to explain an operation in the context of the end result desired of the program.

Lower case subscripts u and e indicate the portion of a word in INUSE containing respectively the upper and lower limits of the corresponding tables. Upper case subscripts L and R designate the left and right half respectively of the words referred to. In general data will be stored right-justified (i.e., in the right half).



encloses a message to be printed out on-line.

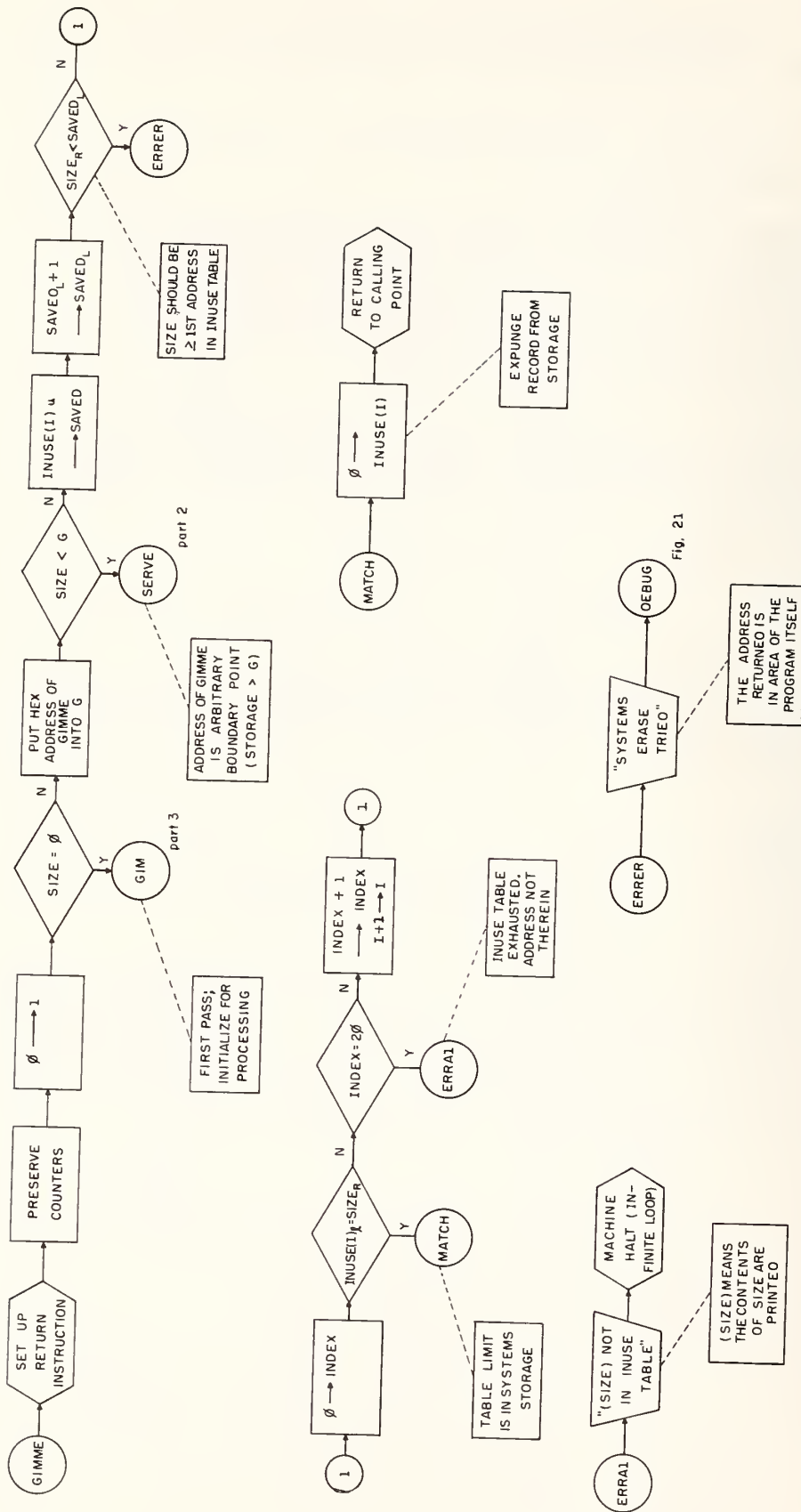


FIGURE 26. FLOW-CHART FOR PSEUDO-DYNAMIC STORAGE ALLOCATION - Part 1

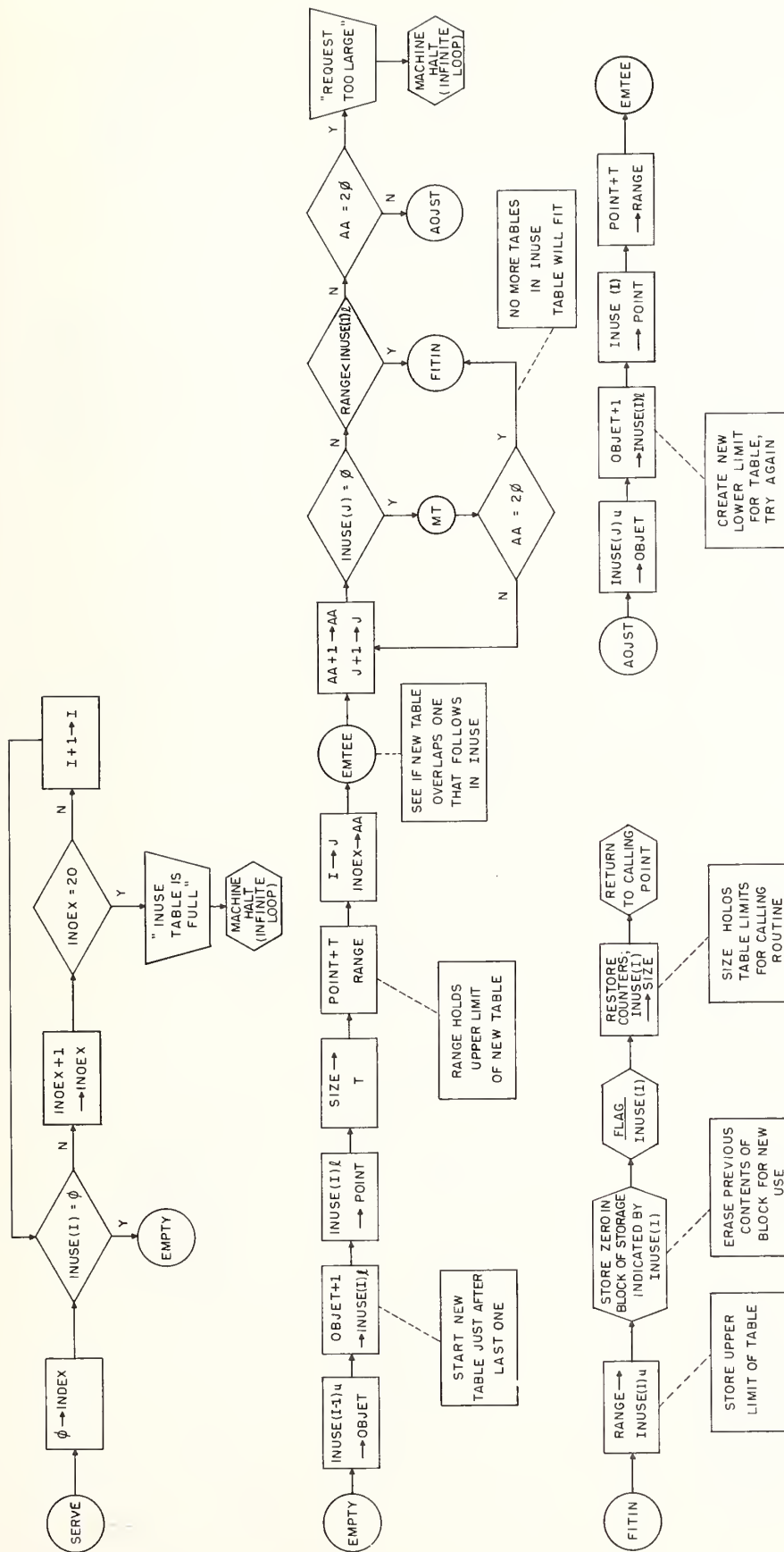


FIGURE 27. FLOW - CHART FOR PSEUDO - DYNAMIC STORAGE ALLOCATION - PART 2

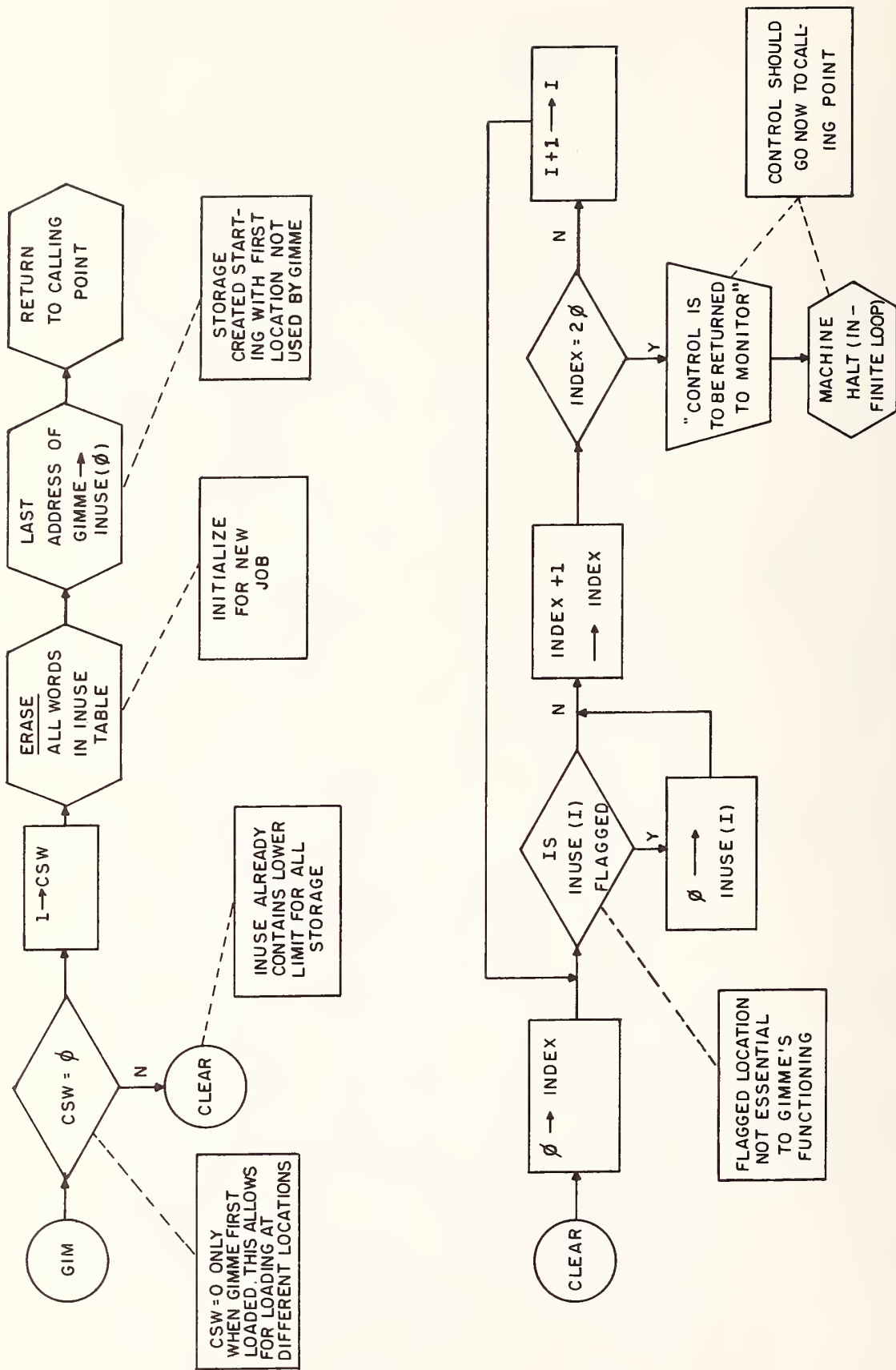


FIGURE 28. FLOW-CHART FOR PSEUDO-DYNAMIC STORAGE ALLOCATION-Part 3

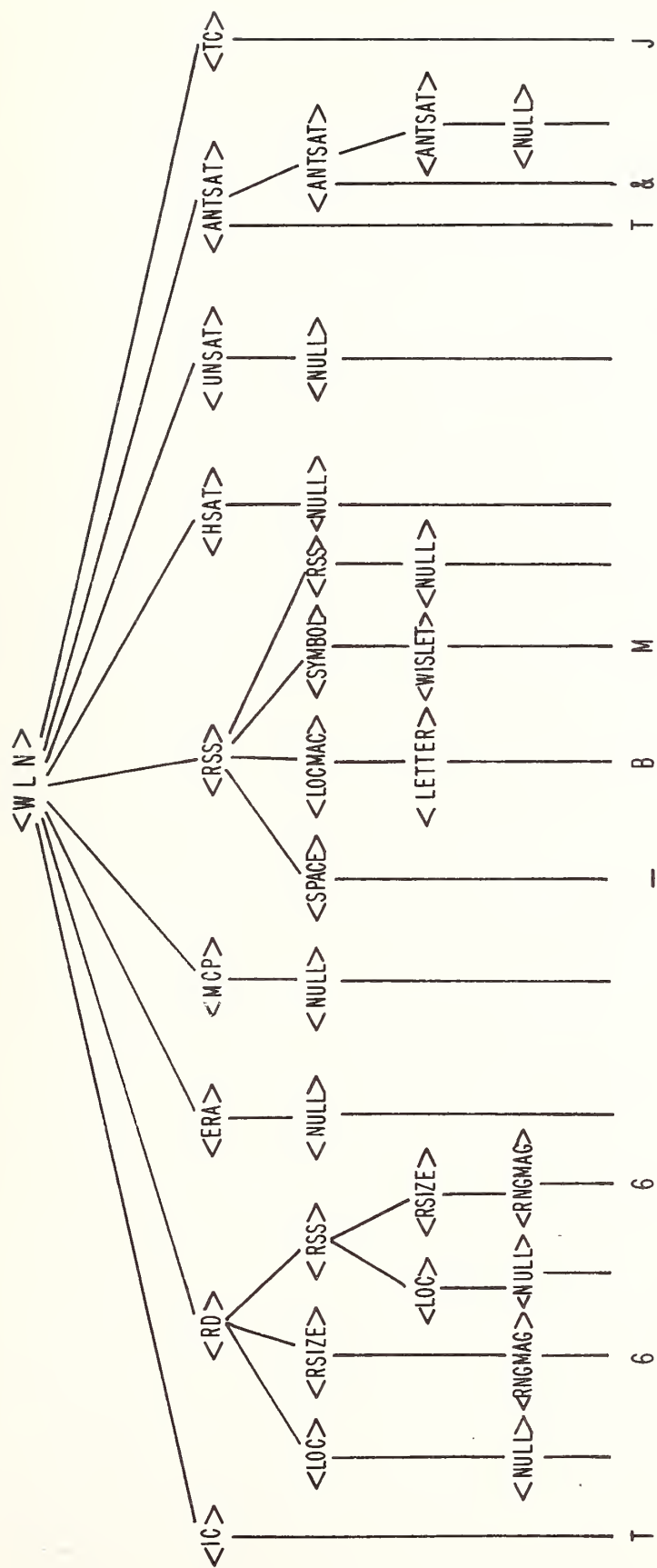


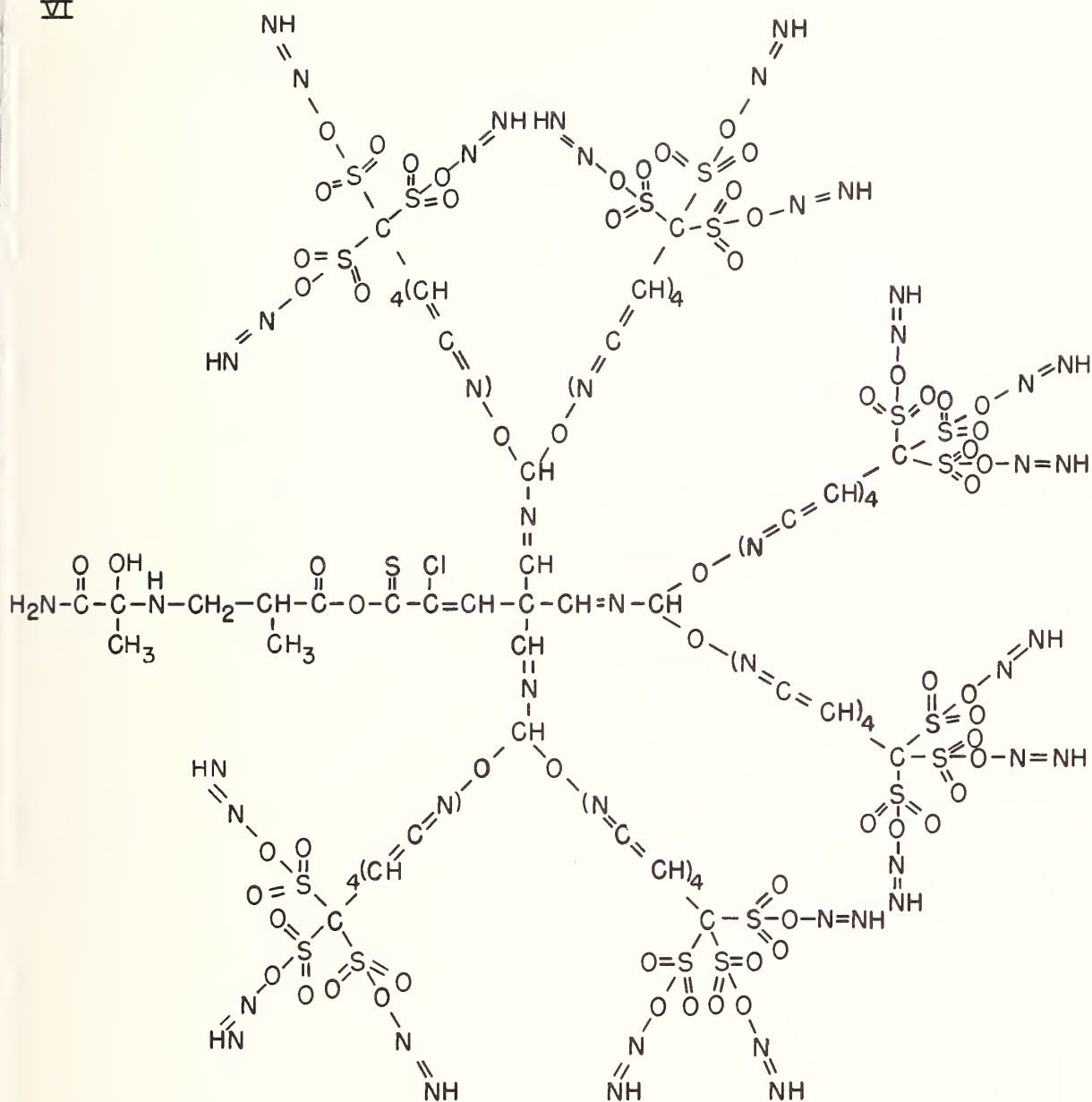
FIGURE 29. PARSING OF THE WISWESSER NOTATION T66-BMT&J

## APPENDIX

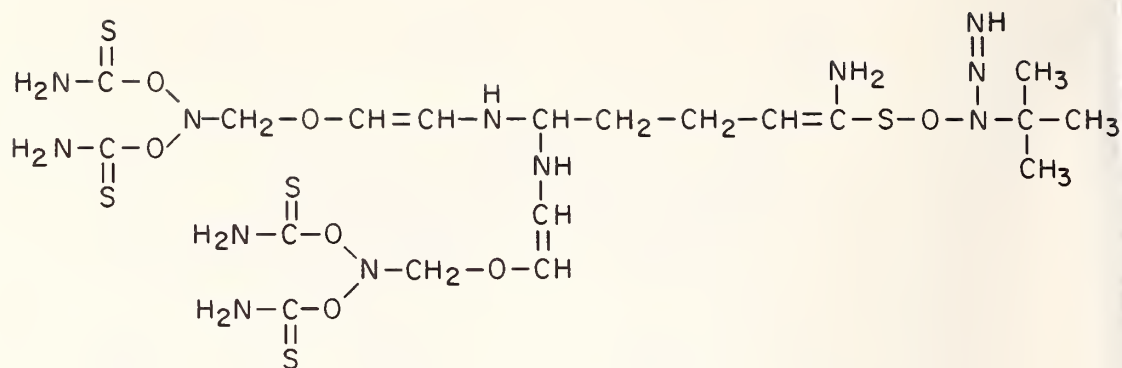
Structure diagrams for Wiswesser Notations

(Roman numerals correspond to those of Notations)

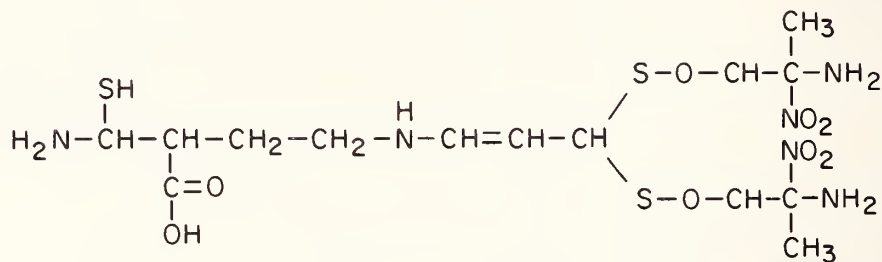




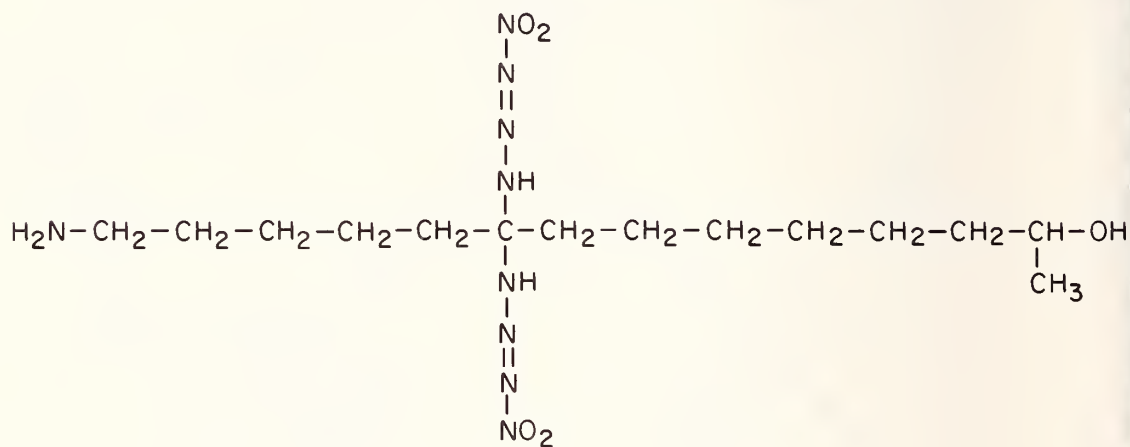
VII



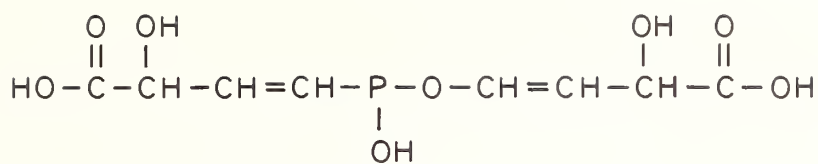
VIII



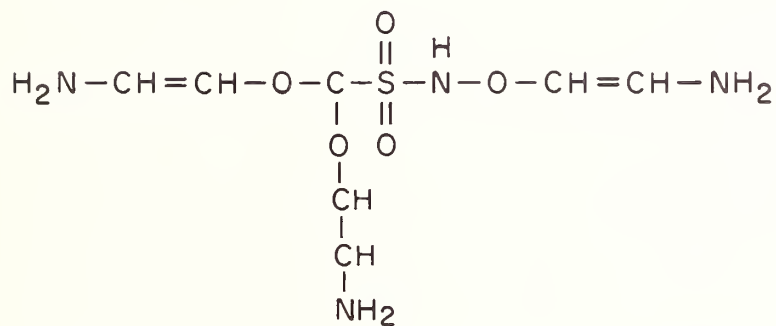
IX



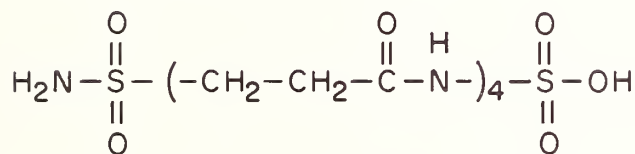
X



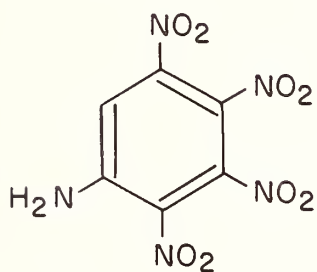
XI



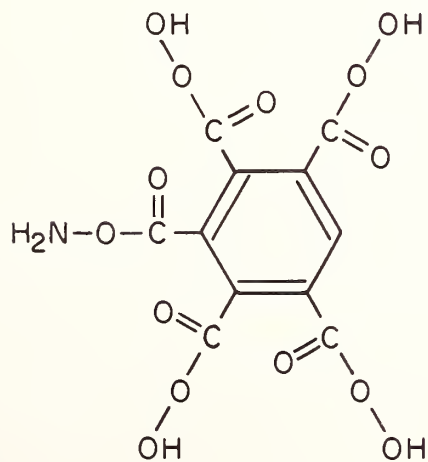
XII



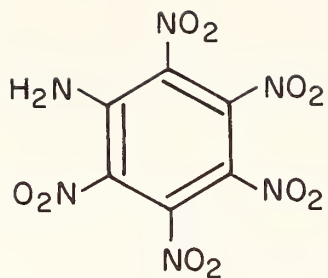
XIII



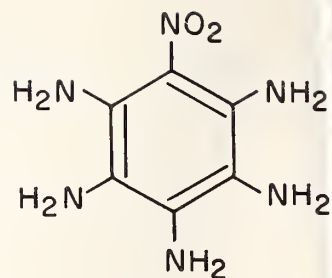
XIV



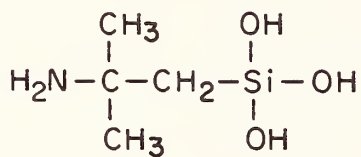
XV



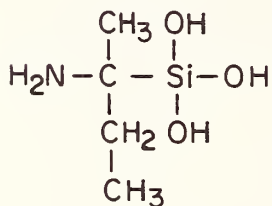
XVI



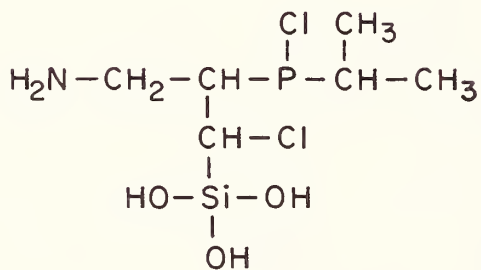
XVIII



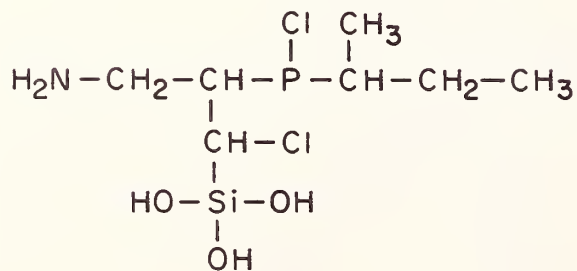
XIX



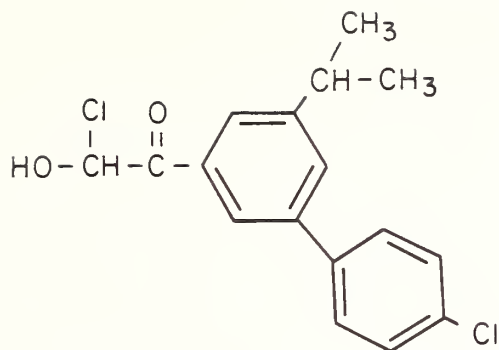
XX



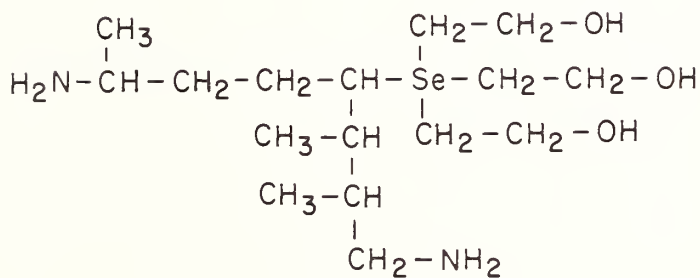
XXI



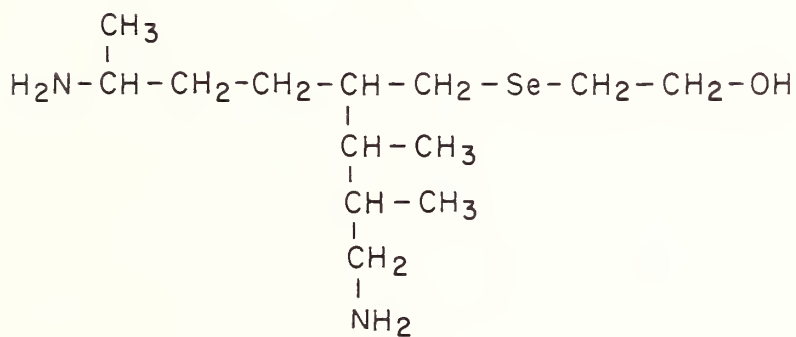
XXII



XXIII



XXIV







# NBS TECHNICAL PUBLICATIONS

## PERIODICALS

**JOURNAL OF RESEARCH** reports National Bureau of Standards research and development in physics, mathematics, chemistry, and engineering. Comprehensive scientific papers give complete details of the work, including laboratory data, experimental procedures, and theoretical and mathematical analyses. Illustrated with photographs, drawings, and charts.

*Published in three sections, available separately:*

### ● Physics and Chemistry

Papers of interest primarily to scientists working in these fields. This section covers a broad range of physical and chemical research, with major emphasis on standards of physical measurement, fundamental constants, and properties of matter. Issued six times a year. Annual subscription: Domestic, \$5.00; foreign, \$6.00\*.

### ● Mathematical Sciences

Studies and compilations designed mainly for the mathematician and theoretical physicist. Topics in mathematical statistics, theory of experiment design, numerical analysis, theoretical physics and chemistry, logical design and programming of computers and computer systems. Short numerical tables. Issued quarterly. Annual subscription: Domestic, \$2.25; foreign, \$2.75\*.

### ● Engineering and Instrumentation

Reporting results of interest chiefly to the engineer and the applied scientist. This section includes many of the new developments in instrumentation resulting from the Bureau's work in physical measurement, data processing, and development of test methods. It will also cover some of the work in acoustics, applied mechanics, building research, and cryogenic engineering. Issued quarterly. Annual subscription: Domestic, \$2.75; foreign, \$3.50\*.

## TECHNICAL NEWS BULLETIN

The best single source of information concerning the Bureau's research, developmental, cooperative and publication activities, this monthly publication is designed for the industry-oriented individual whose daily work involves intimate contact with science and technology—for *engineers, chemists, physicists, research managers, product-development managers, and company executives*. Annual subscription: Domestic, \$1.50; foreign, \$2.25\*.

\*Difference in price is due to extra cost of foreign mailing.

## NONPERIODICALS

**Applied Mathematics Series.** Mathematical tables, manuals, and studies.

**Building Science Series.** Research results, test methods, and performance criteria of building materials, components, systems, and structures.

**Handbooks.** Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

**Special Publications.** Proceedings of NBS conferences, bibliographies, annual reports, wall charts, pamphlets, etc.

**Monographs.** Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

**National Standard Reference Data Series.** NSRDS provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated.

**Product Standards.** Provide requirements for sizes, types, quality and methods for testing various industrial products. These standards are developed cooperatively with interested Government and industry groups and provide the basis for common understanding of product characteristics for both buyers and sellers. Their use is voluntary.

**Technical Notes.** This series consists of communications and reports (covering both other agency and NBS-sponsored work) of limited or transitory interest.

## CLEARINGHOUSE

The Clearinghouse for Federal Scientific and Technical Information, operated by NBS, supplies unclassified information related to Government-generated science and technology in defense, space, atomic energy, and other national programs. For further information on Clearinghouse services, write:

Clearinghouse  
U.S. Department of Commerce  
Springfield, Virginia 22151

Order NBS publications from:  
Superintendent of Documents  
Government Printing Office  
Washington, D.C. 20402

U.S. DEPARTMENT OF COMMERCE  
WASHINGTON, D.C. 20230

POSTAGE AND FEES PAID  
U.S. DEPARTMENT OF COMMERCE

OFFICIAL BUSINESS

---